Implementing Mobile DRM with MPEG-21 and OMA

Silvia Llorente, Jaime Delgado and Xavier Maroñas

Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya
C/ Jordi Girona 1-3, E-08034, Barcelona, Spain

Abstract. Digital Rights Management (DRM) is an important issue when trying to provide advanced multimedia content distribution services. DRM is mostly thought for personal computers, but now, with the emerging mobile devices which include multimedia support and higher bandwidth capabilities, users demand more valuable content for their mobile devices. The most relevant initiatives in the area, MPEG 21 and OMA DRM have been used for implementing a DRM system for mobiles. A lot of work has still to be done in this area and what we present here is a combination of techniques coming from MPEG 21 and OMA for providing a more complex system with combined capabilities.

1 Introduction

The provision of multimedia content for mobile devices is increasing day by day. This implies that content providers want their contents to be protected in a more complex way, too. On the other hand, devices have more resources in terms of screen size, processing capabilities and formats supported and users want to take profit.

In this paper we present a way of providing digital rights management (DRM) for mobiles based on two of the most important initiatives in this area: MPEG-21 and OMA.

The structure of this paper is as follows. First, we introduce MPEG-21 and OMA, and their most relevant parts for the work presented here. Then, in section 4, we describe the DRM system for mobiles implemented. Section 5 presents some commercial implementations of OMA DRM systems. Finally, we present some conclusions in section 6.

2 MPEG-21

MPEG-21 standard deals with different aspects of multimedia information management. In the MPEG-21 context, the information is structured in Digital Items, the unit of distribution and transaction. The other fundamental element are users, which interact with Digital Items. The objective of MPEG-21 is to provide a multimedia framework for assuring that systems providing multimedia content are interoperable.

Llorente S., Delgado J. and Maroñas X. (2007).
Implementing Mobile DRM with MPEG-21 and OMA.
In *Proceedings of the 5th International Workshop on Security in Information Systems*, pages 166-175
DOI: 10.5220/0002428601660175

Apart from the Digital Item, there is a lot of related information that describes additional aspects regarding the content, such as Rights Expression Language expressions (REL, Part 5) [1] and Event Reporting information (ER, Part 15) [2].

2.1 MPEG-21 Rights Expression Language

MPEG 21 REL specifies the syntax and semantics for issuing rights for users to act on multimedia content. One important concept in REL is the License that is a container of grants. Fig. 1 shows the basic structure of a REL License.

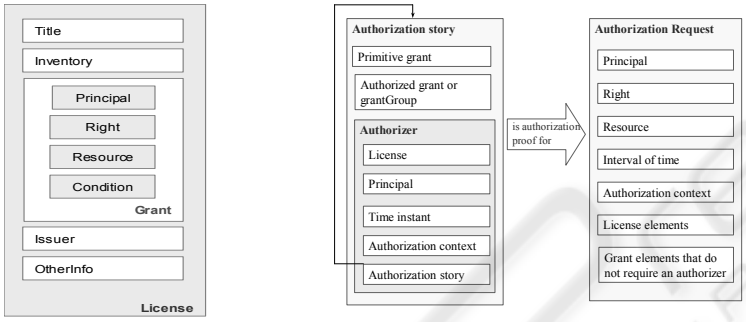

Fig. 1 and 2. Basic MPEG 21 REL License and REL Authorisation Model.

Inside a REL license, the most important element is the Grant, which expresses that some Principal may exercise some Right against some Resource, subject, possibly, to some Condition. A Grant is an XML structure that is formed by four elements: Principal, which represents an entity involved in granting or exercising rights, Right, that specifies an action a principal can perform on a resource, Resource, that represents the object against which the Principal of a Grant has the right to perform, and Condition that represents terms and conditions that must be satisfied to consume the object. The issuer element defines the party issuing this license to the Principal, with the information defined inside the Grant (resource, right and conditions).

Implementation of the MPEG-21 Authorisation model. Inside the MPEG-21 REL standard, an authorisation algorithm based on MPEG-21 REL licenses and some other structures, like Authorisation Request and Story, was defined [1].

The authorisation algorithm defines how a user can be authorised to perform some action over a resource on the basis of the licenses he owns. Moreover, also the chain of licenses is checked, that is, the parent licenses that allowed the creation of the license owned by the user. The authorisation algorithm makes use of an authorization request, an authorization context, an authorization story and an authorizer, as shown in Fig. 2. It allows the definition of different business models depending on distribution. See [3] for details on authorisation of user actions.

Although MPEG-21 defines several structures in its authorisation model, at Distributed Multimedia Applications Group (DMAG) [4], we have defined some UML and relational models [5] in order to represent both MPEG-21 REL and OMA DRM

REL [6] licenses. The aim of these models is to speed up the authorisation of actions based on the licenses owned by a user over a resource. It is also possible to perform translations between these two languages.

When we use both relational and UML models, the authorisation is based on performing SQL queries against a database containing licenses and, from the information retrieved, implement the corresponding part of the Authorisation algorithm. The algorithm checks if the conditions are fulfilled for any of the licenses owned by the user over the resource. To do so, the information retrieved from the database is stored in the classes represented by the UML model and the algorithm works against them, instead of using the original licenses in XML language nor SQL results.

2.2 MPEG-21 Event Reporting

Event Reporting is standardised as part 15 of MPEG-21 standard [2]. With Event Reporting, MPEG-21 tries to provide a standardised mechanism for informing about Events occurred among Peers and Users forming part of a system, which includes several communicating parties. The sending of ER's depends on the system [7].

```
<ER id="ID:mepg:mpeg21:DII:ERDID:002">
      <!-- ======== ER Descriptor ==========-->
      <Descriptor>
            <Format>xml</Format>
            <AccessRights MimeType="text/plain">
                  B: RW, P:RW
            </AccessRights>
            <ERStatus/>
            <PriorityLevel/>
      </Descriptor>
      <!-- ====== Source Descriptor =========-->
      <Source>
            <Peer id="B">B</Peer>
            <OriginalERR
             id="ID:mepg:mpeg21:DII:ERRDID:003">
                  ID:mepg:mpeg21:DII:ERRDID:003
            </OriginalERR>
      </Source>
      <!-- ====== ER Data Descriptor ========-->
      <ERData MimeType="text/plain">
            Broadcaster ID:B, User ID: U,
            DI ID:mpeg:mpeg21:DII:DPDID:001,
            Time:2004-07-15T14:13+1:00
      </ERData>
      <!-- ==== Requested Action Descriptor -->
      <RequestedAction/>
</ER>
```

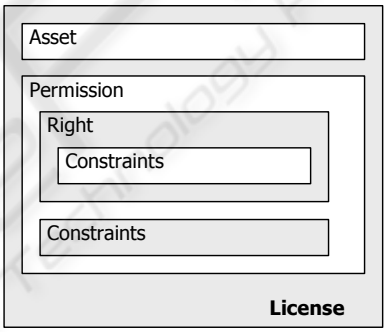

Fig. 3 and 4. Example of an ER and OMA DRM REL License.

In any case, when an ER is generated for informing of a particular action done by a user, it contains information about the associated Digital Item, the resource and the conditions under which this resource was consumed by the user. Fig. 3 shows and example of an ER in XML format.

3 Open Mobile Alliance (OMA) DRM and OMA DRM REL

Open Mobile Alliance (OMA) [6] 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 [6], 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 [6], based on ODRL [8]. It is currently in its version 2. Using OMA DRM REL it is possible to express rights over an asset (content) defining permissions and constraints (conditions) on its usage. To define OMA DRM REL language, ODRL XML Schemas are used [9], which define the rights expression language and the rights data dictionary separately. OMA defines its own rights expression language and data dictionary, both based on ODRL ones.

Although ODRL permits the creation of licenses with a structure very similar to MPEG-21 REL ones, the subset defined by OMA DRM REL is more limited. Fig. 4 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. Moreover, the rights are a subset of the ones in MPEG-21 REL. Moreover, it does not define the issuer, either, as it is usually the company providing the service to the mobile. The constraints can be defined at permission and right levels. Constraints at permission level affect to all rights defined inside the permission.

4 Implementing a Mobile DRM Solution

This section describes the implementation of a mobile DRM solution that selects and combines different mechanisms from both MPEG-21 and OMA DRM, making the necessary adaptations in order to be able to provide a functional solution for the mobile environment on the mediaMobil [10] project. A derivation of some of the modules is also being developed in an Integrated Project, AXMEDIS project [11].

The reason for using both standards is that each of them provides different features suitable for the mobile environment. For instance, we decided to use OMA DRM REL licenses to express the rights given to users of the mobile devices as they were more compact and intended for the mobile environment, but, as authorisation algorithm we have used our adapted version of the MPEG-21 REL one. Moreover, we have used MPEG-21 ER concepts to keep track of user actions, although we are also using a compact version of Event Report information (to store them on the mobile device).

The limitations mostly came from the characteristics of the devices used: limited resources, limited storage and processing, size, etc. Some other limitations where imposed by the development environment available for the mobile devices, as the Java language libraries support depends on the brand and the model of the mobile device.

Fig. 5 shows the general architecture of the system implemented. Several services are involved and different versions had to be implemented. The mobile device has an implementation of the authoriser, part of the Event Reporting and a Status Cache for

storing some information of the actions performed, to be able to authorise them. On the server side, we have two services, the Rights and the Event Reporting Managers.

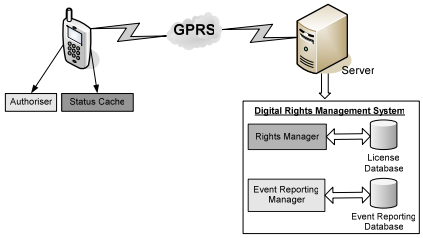

Fig. 5. General Architecture of the Implemented system.

4.1 Licensing

The Licensing implemented in this system works by using Web services and a web application implemented using servlets and Java Server pages. The user of the mobile device is who asks for a new license for consuming multimedia content. To do so, it is needed that he installs a special type of Java application specific for mobile devices called Java Midlet. The implemented application has several menus, which guide the user in the purchasing of a license. In our case, the user should fill three parameters:

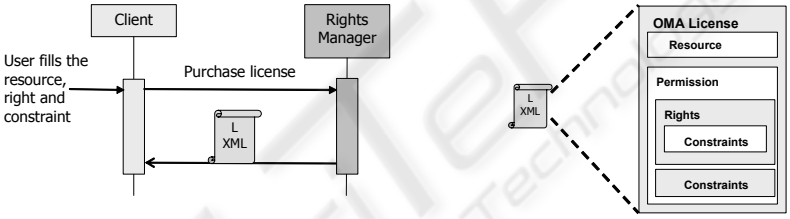

Fig. 6 and 7. License purchasing and Purchased OMA License general structure.

- The resource the user wants to consume with this device.
- The right the user wants to exercise over the resource. The rights available are: Play, Display, Execute, Print and Export. They are defined in OMA DRM REL.
- The constraints for performing the consumption of the resource. We are currently supporting two types of constraints: Time and number of consumptions constraints. For time constraint we can have 1 minute, 1 hour and 1 day constraint. Time constraint indicates that, from the first time the right is exercised, the user has 1 minute, 1 hour or 1 day to exercise the right again, respectively. Status Cache controls the time of resource consumption. For number of consumptions constraint, the user has to indicate how many times he wants to consume the resource. Each time the user exercises a right, it is stored in the Status Cache to control he has not reached the maximum.

Fig. 6 shows the general behaviour of license purchase. The client is the application installed in the mobile device and the Rights Manager is the Web Service based application where licenses can be purchased and managed.

Fig. 7 graphically shows the general format of the license sent to the mobile user. It contains the Resource, the Permission, which contains the Right and the constraints associated to the right and, finally, the constraints for the permission, if any.

```
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
    xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
    xmlns:oma-dd="http://www.openmobilealliance.com/oma-dd"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
    <o-ex:context>
        <o-dd:version>2.0</o-dd:version>
        <o-dd:uid>RightsObjectID</o-dd:uid>
    </o-ex:context>
    <o-ex:agreement>
        <o-ex:asset>
                <o-ex:context>
                    <o-dd:uid>ContentID</o-dd:uid>
                </o-ex:context>
                <o-ex:digest>
                    <ds:DigestMethod
                    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                    <ds:DigestValue>DCFHash</ds:DigestValue>
                </o-ex:digest>
                <ds:KeyInfo>
                    <xenc:EncryptedKey>
                        <xenc:EncryptionMethod
                        Algorithm=
                        "http://www.w3.org/2001/04/xmlenc#kw-aes128"/>
                        <xenc:CipherData>
                            <xenc:CipherValue>
                                EncryptedCEK
                            </xenc:CipherValue>
                        </xenc:CipherData>
                    </xenc:EncryptedKey>
                    <ds:RetrievalMethod URI="REKReference"/>
                </ds:KeyInfo>
        </o-ex:asset>
        <o-ex:permission>
                <o-dd:play>
                    <o-ex:constraint>
                        <o-dd:datetime>
                            <o-dd:end>2007-12-31T23:59:59Z</o-dd:end>
                        </o-dd:datetime>
                    </o-ex:constraint>
                </o-dd:play>
        </o-ex:permission>
    </o-ex:agreement>
</o-ex:rights>
```

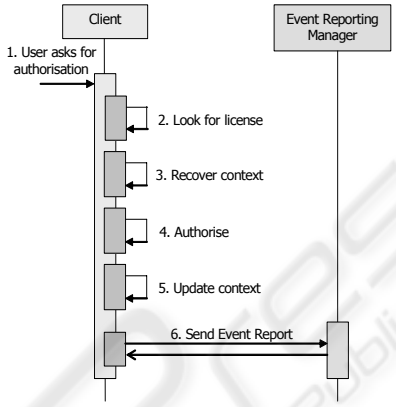

Fig. 8 and 9. OMA License in XML format and Steps for Authorising Content Consumption.

Fig. 8 shows an OMA license in XML format. In this license, the Resource is represented by the identifier ContentID inside the XML element `o-dd:uid` of the element `o-ex:asset`. The right is play, as indicated by the element `o-dd:play`, inside the `o-ex:permission` element. Finally, the constraint represented is a time constraint associated to the right. The element representing it is `o-dd:datetime` inside `o-ex:constraint`. If the constraint type had been number of consumptions, the element representing it would have been `o-dd:count`.

4.2 Authorising

Once the user has purchased the license, he can exercise the right over the resource contained inside it. The user requests the authorisation of the action, indicating which resource he wants to consume and which right he wants to exercise. This request generates several actions that end in the authorisation or not of the exercise of the right.

The steps, shown in Fig. 9, for performing the authorisation are as follows:

1. User asks for authorisation. He has to indicate the resource he wants to consume and the right he wants to exercise.
2. With this information, the client application tries to find out the license which may authorise the requested right in the local storage.
3. Next step is to recover the context information (number of executions, etc), associated to this right and resource from Status Cache.
4. Once the license and the context information associated to the right and resource have been found, the authorisation algorithm can be executed. If the constraints

are accomplished, then the authorisation response will be positive. If not, the next license found will be used for trying the authorisation again. If no license is able to authorise, then the authorisation response is negative.

5. After authorisation, the context information is updated depending on the constraints of the license. If it is of type number, number of consumptions is stored. If it is of type time, the first time when the authorisation was requested is stored.
6. Finally, the Event Reporting Manager is informed of the authorisation request done by the client. This is the only step that requires external connection.

4.3 Keeping Track of User Actions

As explained in section 4.2, to keep track of user actions, an event report is sent after the user requests an authorisation. The event report is only sent on positive authorisation (it is not relevant for the system if the user cannot exercise the right), to limit the number of payment connections done by the mobile device. Positive Event Reports are stored in the Event Report database to be consulted in the future.

We have seen in section 2.2 that the MPEG-21 Event Reports are expressed in XML format. We performed an adaptation of this report format, designing a relational database, instead of working with XML files. The table has 5 different fields, all of type String: EventID, that identifies the Event Report, ObjectID, that identifies object the over action has been done, Action, that for resources indicates the right and TimeofIssuance, that indicates the date and time when the event report was generated.

The client part for requesting the event reporting information was not implemented as the parties that could be interested in where not part of the project.

4.4 Storing Information in the Mobile Device

In order to be able to perform authorisations, two different types of information are locally stored in the device: licenses (in XML format) and context information. Context information was designed as a relational table, but its specific implementation depends on the mobile device system being used. Licenses and context information are both saved into a J2ME RecordStore object. This object lets us to store a series of bytes, indicating only the position where it will be stored. To store something in this structure we have to create a string containing all context information. This means that, when you want to retrieve specific data, you have to go one by one and parse the information, as there is no search functionality provided.

The information inside the context information is the identifier of the license returned by the Rights Manager, the identifier of the resource, the right and, depending on the constraint, a time instant or a number of executions. Context information is essential for authorisation, as it stores the relevant values for the constraints contained in the licenses.

4.5 Problems Encountered using Different Mobile Platforms

During the implementation and set up of this system, we used two different mobile devices, a Qteck 2020i PDA with Pocket PC and a Nokia 6680 mobile phone with Java support. The implementation of the application was done with Netbeans 5.0, using some J2ME support libraries for accessing Web Services and web applications. Those libraries were j2me-ws for connecting with Web services from J2ME and kxml v2 for parsing the license for authorising content consumption requested by the user.

For the Qteck PDA we were able to install the Midlet application with direct access to Web Services. To do so, we included into the Midlet the j2me-ws library, as it was not available in the Pocket PC platform natively. Then, the client application was able directly connect to the server containing Rights and Event Reporting Managers.

On the other hand, for the Nokia mobile phone, we were not able to include j2me-ws library, as the mobile does not support the addition of new libraries. So, if a library needed by a Midlet is not inside the mobile natively, then the Midlet will not work. To overcome this problem, we had to implement a web application that acted as a client of the Web Service. In this case, the mobile application contacted the intermediary web application, which sent the information to the Web Service. The response from the Web Service was passed from the intermediary to the mobile application. The rest of the functionality was the same in both cases, we only had to change the connection with the Web Services due to the limitations of the Nokia phone. The solution to this problem is to use another Nokia phone model which included the required library, but it was not available in the Spanish market at the time when the project was being developed.

4.6 Some Views of the Application

In this section we show some screenshots of the implemented application. It is based on forms for facilitating the user the introduction of information, in order to overcome the difficulty associated to the user of the PDA and mobile phone keyboards.

Fig. 10 shows the main screen of the Java Midlet. The available operations are buying a license with time constraints or number of executions constraints, ask for authorisation of a user action and look for information stored inside the Status Cache.

Fig. 11 and 12 show how a time constraint license can be bought. The resource field has to be filled and the right and the time constraint type have to be selected. In this case, operation has been successful and the identifier of the license bought is shown, LID144.

Fig. 13 shows the authorisation of an action. The user fills the resource name and selects the right. If authorisation is positive, the Event Report number generated is returned. Fig. 14 and 15 show how to request information inside the Status Cache. Resource name and right is needed (similar to authorisation). For the license search, the result shows license identifier and the XML file. For context information associated to a time constraint, the result is the license identifier, the resource name, the right and final time when the right can be exercised.

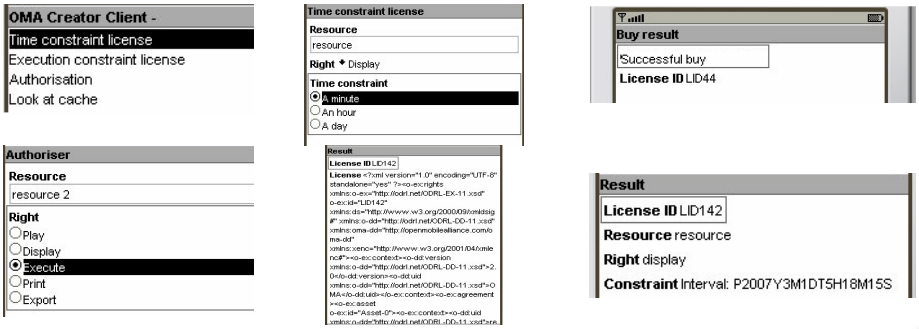

Fig. 10 to 15. Views of the application.

5 Related Work

There are some commercial solutions that implement OMA DRM on mobile devices. In this section we present the most relevant ones and the solutions they propose in the mobile environment. The first one is a client-server model solution that comes from Nokia [12]. They have implemented a system only operational for Nokia products that allows controlling all the process from the content creation until the content consumption. Another commercial solution is the one developed by CoreMedia [13]. It is currently applied to content offered by Vodafone, through Vodafone Live! service [14]. They offer a framework supporting OMA DRM v.1 and v.2 [6] or even Windows Media DRM [15]. They implement something similar to Event Reporting to control user actions. Other companies like Discretix [16] or SafeNet [17], offer equivalent products.

Comparing our implementation with the commercial solutions, we see that we provide two extra features over commercial solutions. The first one is a complete Event Reporting mechanism, based in MPEG-21 standard. The other added value is that our software is not a commercial product but the result of a research project and it is based on open specifications. Our main advantage is that our implementation could be used on any mobile device, independently from the device brand and even from the type of device, permitting interoperability.

6 Conclusions and Future Work

In this paper we have presented a DRM system for mobiles implemented in a research project. This system included different modules that were based on the most relevant initiatives in the area: MPEG-21 and OMA DRM. We have selected some components from both. This is a first step to provide interoperability between them.

Nevertheless, this is only a small part of all what has to be done in order to provide a complete DRM system for the mobile environment. The following (but not only) features should be added in order to provide a secure and reliable system:

- Security in the communication between the mobile device and the services.

- Connection of the DRM features with the player application rendering the content to the user. Content should be also protected (ciphering).
- Ciphering of the Status Cache.
- Use of licenses expressed in MPEG-21 REL.

Most of these functionalities were planned in the system we wanted to implement, but, due to the limitations in support libraries for mobiles and the timing of the project, it was not possible to add them. This is the future work in this area, together with the integration with the OMA DRM v2.0 already present in some mobile devices, like Nokia N95. Nevertheless, integration with the OMA DRM inside this mobile device is only supported by the specific C++ framework provided by Nokia, so a more generic application implemented with Java Midlets will not be able to be integrated.

Acknowledgements

This work has been partly supported by the Spanish Administration (DRM-MM project, TSI 2005-05277), the i2cat Machine project [10], the FP6 Integrated Project AXMEDIS [11] and the FP6 Network of Excellence VISNET-II [18].

References

1. ISO/IEC 21000:5 – Part 5: Rights Expression Language. In ISO/IEC Standards. ISO/IEC.
2. ISO/IEC 21000:15 – Part 15: Event Reporting. In ISO/IEC Standards. ISO/IEC.
3. Rodríguez, E., Llorente, S., Delgado, J. Use of Rights Expression Languages for protecting multimedia information. In WEDELMUSIC 2004 Proceedings. IEEE Computer Soci-ety.
4. Distributed Multimedia Applications Group (DMAG). http://dmag.upf.edu.
5. Llorente, S., Delgado. J., Barrio, R., Maroñas, X. Translation between XML based rights expressions using UML and relational models. In AXMEDIS 2006 Proceedings. IEEE Computer Society.
6. Open Mobile Alliance (OMA). In http://www.openmobilealliance.com.
7. Torres, V., Delgado, J., Llorente, S. An implementation of a trusted and secure DRM architecture. In Lecture Notes in Computer Science 4277. Springer-Verlag.
8. Ianella, R., 2007. Open Digital Rights Language (ODRL). In http://odrl.net.
9. Open Digital Rights Language (ODRL) XML Schemas. In http://odrl.net/1.1/ODRL-EX-11.xsd and http://odrl.net/1.1/ODRL-DD-11.xsd.
10. Fundació i2cat. http://www.i2cat.cat.
11. FP6 AXMEDIS project. http://www.axmedis.org.
12. Nokia development Forum. http://forum.nokia.com.
13. Coremedia. http://www.coremedia.com.
14. Vodafone live. http://live.vodafone.es.
15. Windows Media DRM. http://www.microsoft.com/windows/windowsmedia/for-pros/drm/default.mspx
16. Discretix. http://www.discretix.com.
17. SafeNet, 2007. SafeNet Inc. http://www.safenet-inc.com.
18. FP6 VISNET-II Network of Excellence. http://www.visnet-noe.org