Multimed Tools Appl
DOI 10.1007/s11042-013-1468-y

Exploiting P2P scalability for grant authorization in digital rights management solutions

Pierfrancesco Bellini · Paolo Nesi · Fabio Pazzaglia

Abstract Digital rights management solutions are today quite widespread. Their cost is still quite expensive, and thus in many cases, their application is limited to specific business cases. On the other hand, the market still offers large cases where scalable DRM solutions would find their applicability, for example the management of complex cross media content such as the one used for the educational content, for electronic medical record, etc. In this paper, the focus is on reducing DRM costs by solving scalability problems lying behind the complexity of granting authorizations and performing verification for a large number of users, content and rights associated with them. The proposed solution is based on the exploitation of the DHT P2P network and storage to cope with verification and grant authorization. The paper reports the structure of the DRM solutions, the details for including the DHT P2P into the DRM architecture. The paper also reports details on how the proposed P2P DRM solution can be integrated into traditional DRM solutions. The provided experimental results have proved the reduction of costs, the scalability against the aforementioned cases. The studies and solutions reported in this paper have been worked out and validated on top of MPEG-21/AXMEDIS DRM solutions and tools. On the other hand, the solution is general enough to be adapted in other DRM solutions.

Keywords DRM complexity · DRM P2P · MPEG-21 · DHT · AXMEDIS · Rights controls · Media distribution and protection · Rights control of electronic medical record

1 Introduction

Digital rights management, DRM, solutions are today quite widespread in industrial applications, for example those of: Apple, Microsoft Windows Media DRM, Adobe DRM, and for others see [17]. Some standard solutions have been also defined such as OMA (Open

P. Bellini · P. Nesi (✉) · F. Pazzaglia
DISIT-DSI, Distributed Systems and Internet Technology Lab, Dipartimento di Sistemi e Informatica,
Università degli Studi di Firenze, Via S. Marta, 3, 50139 Firenze, Italy
e-mail: paolo.nesi@unifi.it
URL: http://www.disit.dsi.unifi.it

Published online: 26 April 2013

Mobile Alliance, [15], MPEG-21 [26], MPEG-21/AXMEDIS DRM [12, 13]. Most of the above-mentioned DRM solutions can cope only with specific combinations of media formats, distribution channels, and devices. They have a limited or absent interoperability among one another. In terms of licensing, most of these DRM solutions allow exploiting/controlling only a limited number of rights for each digital content and therefore they allow exploiting only a limited number of business models. For instance, they can control the content playing on the selected platform which the content has been bought for and have no or limited flexibility in controlling the rights regarding editing and changing the content or single elements of complex rich content. The present usage of DRM solutions is quite limited to some business models, while for other applications, the DRM solutions could be the only viable solution to preserve the privacy and control a large number of rights. The standardization efforts (such as MPEG and OMA) focused on defining common frameworks where a larger number of rights maybe defined, MPEG-21 [20, 27], OMA [15, 22]. Among the standards, in terms of flexibility and interoperability, the MPEG-21 is one of the most promising, it can deal with: license (formalized in some REL, Right Expression Language) and the protection information called Intellectual Property Management and Protection (IPMP) [21].

Recently, new challenges are demanding to DRM capabilities to control and manage complex digital content (called cross media and/or rich media content), for example for the Electronic Medical Record (EMR) (typically produced and kept from hospital and doctors) or for Personal/patient Health Record, PHR; and for delivering valuable and complex educational courses, etc. The protection and DRM of complex cross media content has been discussed in [9]. These kinds of content are typically interactive with several types of media interrelated files inside: audio, video, games, document, etc., glued and/or formalized as eBook/ePub, SCORM, MPEG-21, HTML5, MXF, NewsML, etc. Some of them are distributed in some binary format such as ISOMEDIA and/or some ZIP coding as NewsML and ePub, see [12, 13]. Details about PHR and the needed protection models are reported in [14, 16].

Users expect to access and work on this kind of content from different devices according to different licensing/business models (e.g., accessing, renting, pay per play, pay per use, all you can eat, passing the content to friends and getting some bonus for it). The usage of the above described content kinds (such as the EMR or rich media) implies to provide authentication of content elements and secure protection of personal information, and, at the same time, it has to be possible to grant the possibility of adding and changing information to some qualified users. For example, when the doctor gets access to the EMR and needs to add new comments to the analysis results, and/or the description of a disease. Final users are interested in granting some rights to other users for limited time; to a new doctor, to the analyst, etc.; to move the content from one device to another, at home and/or on the street if the content access is performed in emergency events, etc. Typically, both educational packages as well as the EMRs are constituted by several files and text (maybe produced by several actors, different teachers, different hospitals), which in turn can be augmented with other files and annotation in some conditions by authorized personnel only. At the same time, some elements of the rich content should be accessible for a limited number of authorized people; the organization can enrich the EMR automatically and provide updates to all the authorized people, etc. Therefore, the rich cross media content should cope with different rights related to both different parts/elements and the whole package. Therefore, the package is evolving over time, while some copies should evolve as well, together with the main copy. The protection of the complex rich media may be encrypted with different keys on different files. Set of files can be located as nested levels into the encryption model and in agreement to the protection information and coding [13].

Therefore, in the above described cases, the number of rights per content can be high, and thus the challenging scalability problem of DRM solutions has to manage content on demand at a reasonable cost. Thus, the solution has to cope with hundreds millions of users, several millions of different content items, to control a large set of possible rights for each content and user (i.e., adapt, play, access, embed, export), on different devices, etc. Most of the present DRM solutions cope with complexity and management costs settling such aspects by a compromise on the number of users for service, and/or on the numbers of content per service, and/or on the number of rights per content and thus of licenses, on the limited kinds of business models, on the security level and robustness with respect to the attacks, etc. The adopted business model strongly influences the cost of the solutions, since it may relax some constrains regarding the verification of the devices, the authentication of the users, the production of grant authorization at each transaction; for example when licenses storage on the final user devices is permitted.

Typically, rights are formalized as logic constraints in licenses which in turn have to be processed as logical rules in order to compute if a given user may have the grant to exercise his right on a given content at a given time and in a given location, etc. Licenses and logic constraints defining rights can be connected one another with chains [21]. For example, when a protected content package includes several other content packages with different rights, or when a license for distribution allows the distributor to issue licenses for playing to final user customers, etc. In any case, the proliferation of rights to be granted and controlled on the device implies to cope with huge amount of licenses to be stored, distributed and processed in real time (on demand). For example, (i) when millions of people get access to the same sport event content in short time; and/or (ii) when millions of people are accessing content with different authorizations and from different locations and devices. In these cases, the complexity of the problem is mainly on the information related to the grant authorization (rights), which may imply the corresponding computation of license logic, but only after the verification of the user/device's identity, that is the verification/certification process.

The complexity and costs for granting authorization according to some license is typically demanded to a single central service that has to provide: (i) suitable storage for the information, (ii) computational capabilities to compute the logical rules, and (iii) networking capabilities to satisfy millions of users requesting the grant authorization at almost the same time. On the other hand, if the workload during the rush hours has to be met, the hardware infrastructure has to be dimensioned for the worst cases, with corresponding costs.

In this paper, the focus is on reducing DRM costs by solving the scalability problems behind the complexity of granting authorizations and on performing verification for a large number of users, content and rights associated with them. The studies and the solutions reported in this paper have been worked out and validated on top of MPEG-21/AXMEDIS DRM solutions and tools [10, 11]. AXMEDIS has been defined as a consortium of leading European digital content producers, integrators, aggregators, and distributors, together with information technology companies and research groups (see http://www.axmedis.org). The proposed solution is based on the exploitation of a DHT P2P network and storage to cope with verification and grant authorization.

In literature, the usage of P2P network with DRM has been already addressed to store protected objects [1, 7]. [7] has described a P2P network for content distribution where the DRM protected content can be distributed. In that case, the P2P network is only used to share content and not verification and licensing information. The solution presented in this paper complements the solution proposed in [7] which was limited to the exploitation of P2P for content sharing and distribution. In [1–3], the P2P network based on DHT has been used to index rights and metadata allowing users to search content integrated with licenses (the so called governed resources) by looking for specific license issuers, right grants or principals (right recipient user) in the P2P. This last case of iDRM [3], the proposed P2P solution presents unsolved complexities in the DRM management, since the verification and authentication activities are still performed by referring to a central server, without exploiting the P2P network. Moreover, the iDRM solution presented a limited flexibility since licenses are supposed to be simple and independent one another, in any case managed in the license server.

On the other hand, the solution proposed in this paper can cope with (i) the user and device authentication, (ii) flexible licensing including groups of people and devices (i.e., the domains), and licenses with temporal and consumption constraints (i.e., a licenses valid for a month, a license valid for 4 plays), and with licenses depending on other licenses; More details will be provided later in the paper. Therefore, the proposed solution is general enough to be adapted by many different DRM solutions and it has been assessed in terms of performance against critical scenarios. To this end, the paper is organized as follows.

- Section 2 offers a short introduction to the DRM Scenarios for content distribution. It is used to highlight main entities, services and relationships of a DRM solution for content distribution and rights management.
- Section 3 describes the AXMEDIS DRM solution and services in terms of service graphs and action diagrams. This allows stressing the points where a DRM solution can be adapted to exploit the P2P capabilities, so as to reduce the complexity and costs of the DRM service, considering scenarios related to the user verification and grant authorization from licenses.
- Section 4 presents the details regarding the insertion of the DHT P2P into the DRM solutions for managing the grant authorizations, and not for managing content as in [7], and [1, 3]. The proposed solution is general enough to be adopted in other DRM solutions since the information stored into the P2P refers to the grant authorization and to the certification of device which are suitable for almost all DRM solutions.
- In Section 5, the performance analysis of the DRM P2P solution is presented. To this aim, the declination of the DHT P2P into the specific cases is described by stressing the most relevant parameters and features that have been tuned during the simulations and experiments.
- Section 5.1 reports the experimental results that allowed highlighting the parameters' values of the DHT P2P which are suitable for two applicative scenarios: the distribution of media content, and the management of security for EMR/PHR. The work presented on this paper does not claim to be an exhaustive analysis of the security and protection problems of EMR, but only stresses that DRM P2P can be one of the needed technologies to reduce DRM complexity and costs.
- Conclusions are drawn in Section 6.

2 Digital rights management scenarios

At present, there is a number of content format packages that may include several basic digital resources (documents, video, images, audio, 3D, animations, etc.) such as: MXF, AXMEDIS / MPEG-21 [21], SCORM [19], ePub, NewsML, etc., that could be used as DRMed packages for containing EMR and/or learning content. Packages can wrap digital resources with other related information (e.g., metadata, identification codes) so as to make them ready for delivery and access right control. Such solutions should be more flexible, if compared with proprietary solutions where the DRM is only applied to the single resources. In fact, in those cases, the single element and as well as the packages may have distinct protection information (encryption models and tools) and rights/licenses, maybe issued by different institutions / industries.

On the other hand, a general view of DRM solutions is needed to understand the main mechanisms as reported in this section. Thus, the typical content production and distribution scenario, which synthesizes the most relevant phases from content packaging to content distribution and licensing, is shown in Fig. 1.

The Distributor arranges a contract/agreement with the Content Producer (the value chain is more articulated, but the details are not relevant in this context). The Distributor may create and protect the produced content package to keep a certain level of control about the exploitation of rights (the hospital or a doctor can play the role of distributor). The Distributor may make business by allowing final-users/consumers to exploit specific rights on content (e.g., 'play', 'print', 'adapt' which typically includes 'modify'). The provided rights, for a given final user, are formalized as logical rules into the license (L, in the figure), such as: authorize user UserID to play content ContentID for D days/months, etc., or to print content twice or more times, or to adapt them only in Germany, etc.

In order to enforce protection, the digital resources are packaged for the distribution by using some encryption algorithms. The Protection Information (P, in the figure, includes key and algorithm ID for decryption of the content) has to reach the final user device in order to allow the player/device to unprotect/open the digital resource and/or the package, but only when it is authorized according to the license, and only after verifying the security of the device and the authentication of the user/device. The information for content decryption is typically called Protection Information or IPMP as in MPEG-21 [21]. In order to permit the exploitation of rights formalized in the license (L) only, the player/device has to verify if it can be authorized according to the license, for example by contacting the License Server. In this first case, the License Server processes the license database to verify if the Authorization can be granted to that specific player, device and user. Alternatively, as a second case, a copy

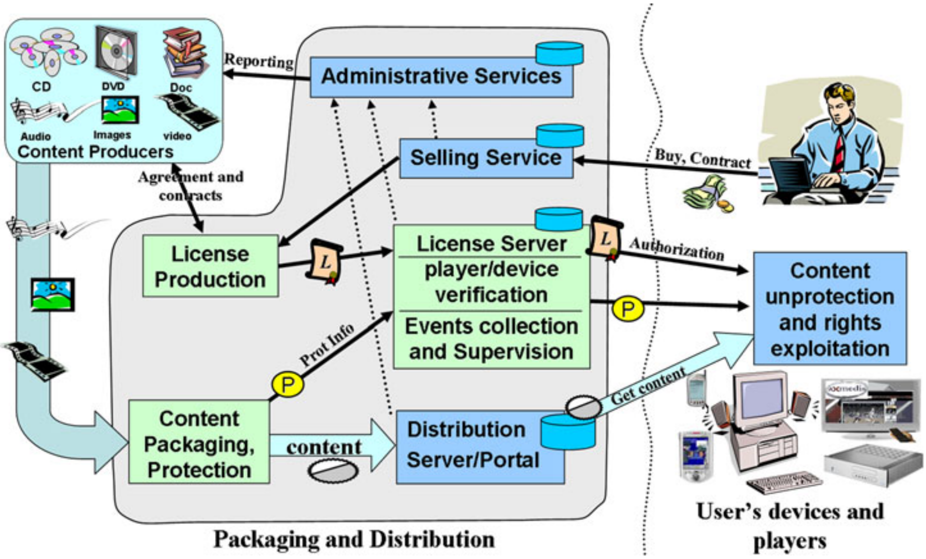
CD
DVD
Doc
Audio
Images
video
Content Producers
Reporting
Administrative Services
Selling Service
Buy, Contract
Agreement and contracts
License Production
L
License Server
player/device verification
Events collection and Supervision
L
Authorization
P
Content unprotection and rights exploitation
Prot Info
P
Content Packaging, Protection
content
Distribution Server/Portal
Get content
Packaging and Distribution
User's devices and players

Fig. 1 Basic DRM: content production/protection and licensing

of the License could be sent to the player only once, when the grant authorization is requested.

Every time a right is exploited, the involved distributors/producers may need to get back an evidence for their Administrative Services (for example, bill sending). This piece of information can be easily recovered in solutions forcing the player/device to contact the License Server for each grant authorization. In those cases, an Action Log record (Event Report in MPEG-21) can be issued by the player/device and reported/saved on some secure server. The information on the user Action Log record can be used for statistics about content usage, to adjust the service, to make market analysis, to monitor the security and it may help in tallying possible attacks.

On the user's player/device, both authorization/license (L), and protection pieces of information (P) are needed to exploit the rights on the content. They can reach the content together or in separate phases.

On such grounds, as to the EMR management, the possibility of keeping separate protection information and authorization is important, since new licensing should be released by users and/or institutions, for a given temporary period, etc., with no need to change the protection coding of the content. For example, allowing a special consultation, a given need to add new information, etc.

2.1 Computational complexity of DRM scenarios

On the basis of the above description, it is evident that the package/content formats, the license, the protection information, the protocols, and the authentication/verification processes exert a significant influence on the DRM architecture and the complexity of the distribution channel, and thus on the workload assigned to the Licensing and Verification Servers. The DRM distribution complexity is related to the number of protected content, number of users, number of licenses for each user and content, the maximum number of requests which may arrive at the same time on the servers. In further details, we can analyze some typical cases.

In the event of content on demand, with millions of users per millions of content, in the worst case, we have: an order or 10^{12} grants/verification to be provided. On the other hand, in a more realistic case: some millions of users may request the authorization in the same short time window, when they decide to acquire/pay a new content via some content on demand service (the access at the same sport event, premier primetime film). In this case, the content is typically protected once, and accessed by millions of users which the grant is provided for on a singular basis and for a single play, which means to play it for a few hours. In this case, the above DRM solution is not very efficient and thus high costs on the licensing servers have to be afforded to guarantee the computational capability, the network bandwidth. The same user, typically does not request the same authorization several times, it typically requests every time different content items and use them only once. For this case, the conditional access systems, CAS, adopted by the broadcasters, have a lower complexity since the Protection Information is broadcasted to proprietary devices that are locally verified by using a smartcard. This approach refers to certified and typically embedded devices, and does not take into account an open player on PC to be verified with respect to the Verification/Licensing Servers. The CAS solutions are not viable for the analyzed case of EMR or for multiple accesses to the same content, as it happens with educational purposes.

In the event of content distribution like Apple iTune and Adobe, millions of users select a limited number of content in life span among the provided range of some million content items. The content is licensed to users forever or for a long period; thus the same content is played/accessed by the same user several times on his/her own certified device. Therefore, on a daily basis only some users perform a transaction to get the first access and thus the license and protection information to the stored into the device. As to Apple iTune or Adobe, the authorization is provided only in the event of first access (thus reducing costs on the Licensing Server). Therefore at the first access/authorization, the license and protection information is stored (or cached) into the device, bringing about a reduction of security. This case is a clear compromise to reduce complexity at the expense of security. On the other hand, this solution provides users with content access at a reasonable price, granting also the possibility of exploiting rights off line (namely, with no connection).

When coming to rich media as EMR, a high level of security should be guaranteed and for this reason the caching of license/rights and/or protection information should not be allowed. In the sub-case of rich media and cross media distribution for educational purposes, the security level can be lower. As to EMR, in a country with 80 million users (which is the population with social security number in Germany), every day, a million grant authorizations to access content may be needed and thus hundred thousands of new licenses should be issued by users, doctors, etc. This increases the complexity of the information to be processed by the License Server. For example, a European doctor may have an average of 280 patients, whereas the same number could be higher for doctors living in other countries. Each person may have an average of 3 doctors (typically 2000 people per dentist, 1000 per pharmacist, 500 per pathologist), this means to have at least 240 million stable licenses being present on the License Server, plus those produced for temporary accesses by specialists, hospitals, analysis cabinets, etc. Moreover, the rich media modeling of the patient health record may be made of several content items with their corresponding licenses and rights (to protect, print, access, ..), protection model, encryption, etc. Moreover, an EMR may be made of 10 files/elements, each of them produced by different Hospital/Doctor and thus they have different licenses. A complexity for storage of about a *number of doctors per patience*number of elements per content*number of rights per element,* thus billions of rights.

As to the grant authorization, the above described DRM system has a linear complexity in terms of maximum number of verifications and authorizations that can be served per day and/or almost simultaneously. Where "served" means to receive the request, retrieve the correct license, compute the license, and provide the results. Moreover, the algorithm for computing and providing the authorizations/verifications has to produce results in real time, this implies high expenses for the infrastructure, including computational and bandwidth costs.

In further details, the license management and the provision of the related grant authorizations cope with:

1. **License production and saving** on the License Server, for all users and rights. Thus, database space and query resources are consumed;
2. **Solving the grant authorization requests** which arrive from both players and the network (both network and CPU capabilities are stressed), implying to perform in real time the following activities:

    - (i) receiving the request for grant authorization, for a specific *user*, about a *content*, to exploit a *right*;
    - (ii) verification of the player/device with respect to the data contained in the central database of million device descriptors (as hardware and software fingerprints) – e.g., each user may have multiple devices, PC, TV, mobile, etc.;

- (iii) estimating grant authorization which consists in searching and processing the returned license(s) (one or more of them associated with the same content, user, right). The search should be performed on billions of records, while the computation is typically limited to a few of them; The phase of processing licenses for computing the grant authorization can be performed only once and the result is stored in some cache;
- (iv) communicating grant authorization results to the requester.

The above discussion on grant authorization phases highlights that among the four phases the most expensive ones are those related to the verification (i), (ii) and to the communication (iv). Phase (iii) has a computational complexity depending on the database costs for retrieving the licenses/rights, and can be reduced by pre-computing and caching them. The storage of licenses cannot be avoided and the present database solutions are much faster than communications.

On the basis of this analysis, the main idea of the solution presented in this paper consists in the adoption a P2P network among final users' players and distributors' servers to reduce the communication complexity and costs for: (i) device/player verification, (ii) grant authorization provision. The proposed solution can be regarded as an affordable solution to transform a centralized DRM architecture in a fault tolerant and cheaper solution in terms of infrastructure and connection costs since a large part of connections and processing activities are delegated to the P2P network. The proposed solution is viable to be used in large scale problems as it occurs with cross media content where (i) several users are engaged in accessing licenses, etc., like in educational content and EMR/PHR (for example, doctors and patients are interested in licensing the access to other doctors and/or family members); (ii) a number of users may be interested in having access to a limited number of content (for example, a family restricted view of the EMR).

In the next sections, the above described problems are analyzed in details to stress the complexity of DRM management and where the P2P solution can be integrated.

3 AXMEDIS DRM solution and services

In this section, the most important aspects of the AXMEDIS MPEG-21 DRM solutions are reported to highlight the device and tools involved from production up to the client side. This section lays the bases for an explanation on the needed changes to introduce the P2P capabilities into DRM solutions. A general description of the AXMEDIS architecture can be recovered from [11] and the full specification from the AXMEDIS web site http://www.axmedis.org. The DRM solution is presented while highlighting the mechanisms for grant authorization, verification / authentication of devices, and license production. On the other hand, even if the solution proposed has been designed and developed for AXMEDIS MPEG-21 solution, it can be applied to other DRM solutions since providing grant authorizations is core part of any DRM solution [26].

In the AXMEDIS MPEG-21 DRM, the role of the License Server is played by the **AXMEDIS PMS Service** (Protection Manager Support). It allows any storing of licenses and authorization granting to the device/players. It is also the primary access for content producers and distributors since it provides services for license store and verification, and also support to revoke licenses, content, tools and users. Some of these services are provided by **AXCS** (AXMEDIS Certifier and Supervisor) and conveniently made accessible via the PMS which establishes the secure communication channel between the Production Tools and Final User's Tools (players/devices). Most services exchange sensitive information (such as certificates, protection information, grants, action logs, etc.) which has to be transferred along a secure channel in order to avoid simple sniffing attacks. The **AXCS** provides services such as: content object ID assignment, user ID assignment, access to Action Log reports and statistics about usage data. The AXCS collects and keeps the information concerning the registered objects, users, devices, etc., and therefore it allows the management of black lists. It also stores the Protection Information of each protected object, and the list of actions performed on them, the so called Action Log database, which allows production of reports and statistics.

The **User Registration Service** allows the registration of users (final and business) in collaboration with the AXCS and the AXMEDIS Certification Authority, CA. The registration portal allows the collection of information about users, regardless of their unique ID's assignment and it provides certificates.

The models and tools mentioned in the scenarios are differently exploited by the different devices and actors of the value chain. For example, "*Producer*" and "*Distributor*" have to be understood as roles rather than effective entities. A User may become producer of a content and then distributor of the same content. The tools mentioned above refer to generic functionalities, as described below. The main scenarios are reported hereafter according to the AXMEDIS tools and web services, in the context of two main scenarios, namely production towards distribution and distribution towards usage. And, in particular for:

- **Production to Distribution** means the activities of (i) protecting content, (ii) distributing content to one or more distributors, (iii) licensing to the distributor the rights to issue licenses to final users, so as to allow them accessing the content. For example, issuing a license with certain rights to a distributor so that it can issue licenses with the same rights to final users.
- **Distribution and Usage** means the activities of content distribution by producing specific licenses for rights exploitation (e.g., content access, print) to final users.

3.1 Activities of producers with respect to distributors

In Fig. 2, the role of the Producers (Content Producer and Protection Tools) with their Production Tools and two related Distributors (see Distribution Services), is presented. The Producer may perform the following operations with respect to the above mentioned services and tools:

- ***User Registration*** for content producers and distributors. The registration may provide them with a unique production ID and release a certificate from the Certificate Authority.
- ***Tool Registration*** to register the tools used by the Producer for producing content object packages, licenses, and to revoke licenses. In general, different users may use the same production tool, so that the certificate is managed at the User's level, whether this distinction is needed by the control and/or revocation process and tools.
- ***User and Tool Authentication*** is performed to authenticate the user and tool every time a connection with the PMS is performed. For example before requesting the grant authorization, posting a license, revoking a license, getting action logs, getting protection information, etc. The authentication is based on certificates which, by using the stored key, allow the building of a protected channel to exchange sensible information such as: user information, device fingerprint (a set of hardware and software identification numbers). A simplification can be performed when User and Tools are considered a unique element, as it occurs with the decoder and the user's smartcard.

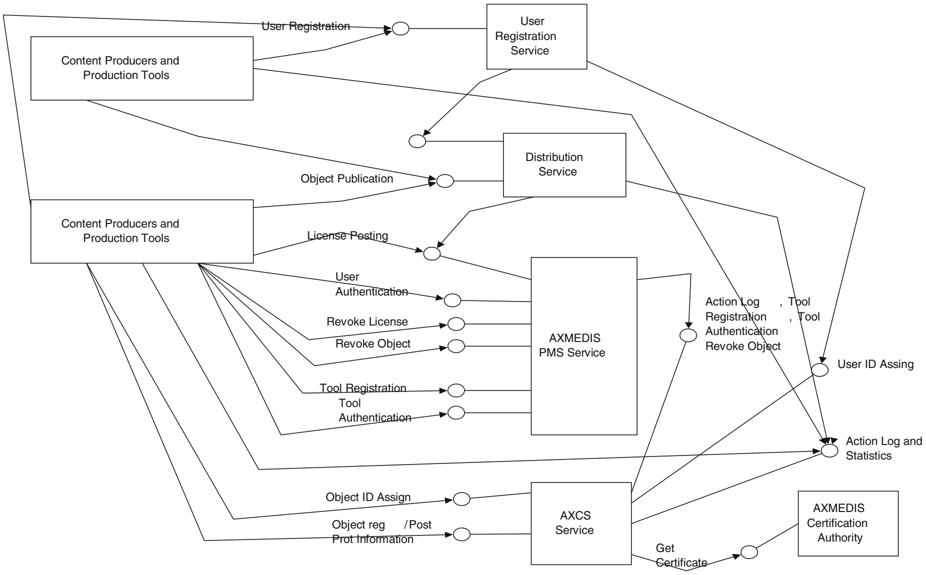
User Registration
User Registration Service
Content Producers and Production Tools
Distribution Service
Object Publication
Content Producers and Production Tools
License Posting
User Authentication
Revoke License
Revoke Object
Tool Registration
Tool Authentication
AXMEDIS PMS Service
Action Log , Tool Registration , Tool Authentication Revoke Object
User ID Assing
Action Log and Statistics
Object ID Assign
Object reg /Post Prot Information
AXCS Service
Get Certificate
AXMEDIS Certification Authority

Fig. 2 DRM Services: point of view of Producers towards Distributors

- ***Object ID Assignment, Object Registration and Protection Information Posting*** operations are performed every time a new content object is published and protected. The Producer registers the object, by requesting an Object unique ID. Then the content can be protected and registered, also providing the Protection Information (i.e., IPMP settings of MPEG-21, protection tools and specific key(s) used) and in some cases the metadata.
- ***License Posting*** is the activity done by the Producer/Distributor (they are license issuers, as well) in formalizing the license including rights granted to the Distributor/Final Users. The Distributor licenses typically include the possibility of: (i) content reselling (by producing final user licenses) according to different business models, (ii) content adaptation to make it suitable for different distribution channels and devices, (iii) content encapsulation (meaning, to include it into an object in other collections, and into an already in place patient health record), etc. Therefore, the Distributor may produce and post on PMS only licenses according to the license parameters defined by the Producer. Similarly, a license issuer may revoke the produced licenses, thus removing the license from the PMS. Only the Producer may revoke the content from the DRM system, putting the Object ID in the black list of AXCS Service, thus removing the possibility of managing, posting licenses, providing protection information and granting authorizations for it.

3.2 Activities of distributors with respect to final users

Figure 3 depicts the point of view of a DRM Client and Player (a DRM enabled tool) with respect to the distribution of content performed by two Distributors and other DRM services. In this case, the actions of a final User are performed by means of the DRM Client and Player. The User may have one or more devices/tools. The DRM Client/player typically carries out the following operations in addition to the User Registration, Tool Registration,

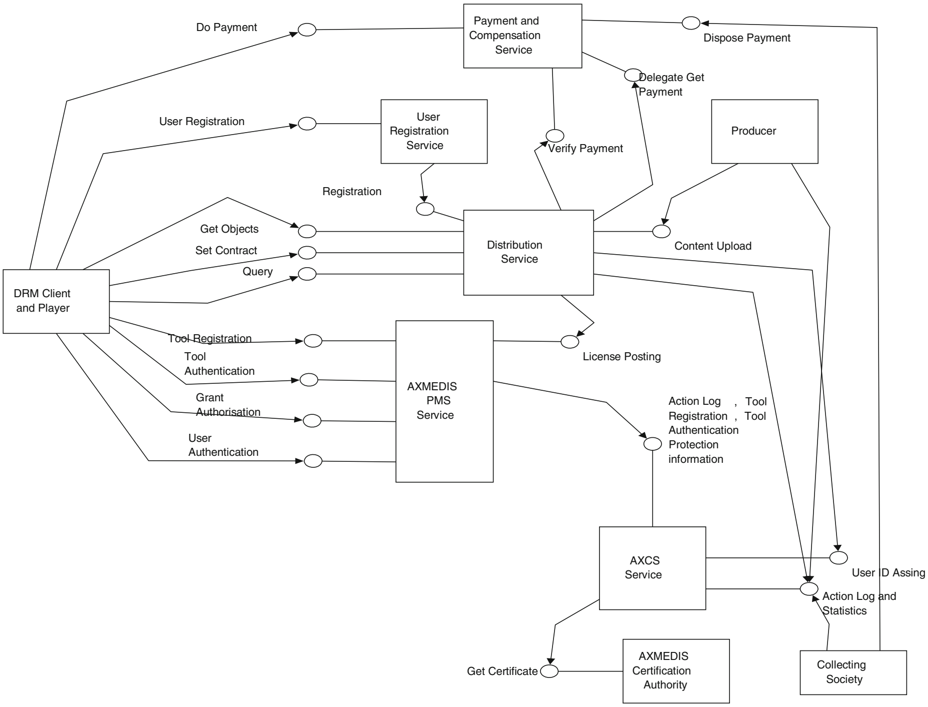

Fig. 3 DRM Services: point of view of final Users from their DRM client and Player, into the user device

User and Tool Authentication, Grant Authorization, which have been already described in Section 3.1:

- ***Grant Authorization*** is requested by a DRM enabled tool registered in advance. The grant authorization is requested to the PMS, and, at the connection with the PMS, the tool and the user are authenticated and verified via their certificate and fingerprints. If the authentication/verification is positive, the grant request can be processed by the PMS which in turn contact the AXCS to get information about tool ID and status, User ID and status, etc. If no problems arise, the license is processed and in the case of matching from the requested grant and the available licenses, the grant is provided. Once the grant is provided, the corresponding action is communicated to the AXCS in the form of Action Log record.
- ***Query, Get Object and Set Contract.*** The final user may go to any Distributor to make some query for selecting the content objects. In order to enjoy the content object, some rights have to be acquired by the user setting a "*contract*" with the Distributor according to some business models (e.g., monthly rate, pay per play, single access limited in time). According to the established *contract*, the Distributor can issue a corresponding License and post it to PMS for that user.
- ***Do Payment.*** According to the business model, the Distributor starts the process to obtain the payment from the user. In the case of patient health record, this feature has no sense. The Distributor periodically verifies the status of the payment to enable the service to its client.

4 DRM with P2P for managing grant authorization process

On the basis of the above presented complexity analysis for the DRM solutions, it is evident that the most critical aspects in terms of complexity / costs focus on to the grant authorization requests and related verifications. The request arrives from the tools/players to the PMS via the network with the aim of performing in real time the following steps:

1. **opening a protected channel** by using the available user/tool certificates and related passwords;
2. **receiving and interpreting the request for grant authorization** (user ID, content ID, tool ID, requested right and related constraints);
3. **verification / authentication of player/device** on the basis of an ***Enabling Code*** estimated on the basis of (i) the player/device fingerprint provided during the first certification/registration of the device/player, and of (ii) the actions performed on the player itself (e.g., play, access, print, etc.) over time. This code is used to perform the verification process, perform the player verification, and thus enabling the player to work in that context and according to its fingerprint and story of the performed actions. Therefore, the right code has to be provided by the device to the PMS. If it is not provided the connection is interrupted and the verification / authentication aborted;
4. **estimation of the grant authorization,** that consists in searching and processing the license(s) associated with the user, for that content, in the period, according to the right for which the grant has been requested. One or more licenses could be recovered to estimate the grant. For example, (i) the distribution (mother) license issued by the producer, authorizing the distributor to issue the license for pay per play; (ii) the license issued by the distributor to formalize the issued right to the final user. A simplification of this process, can be obtained by directly storing if the requested grant can be provided or not according to a given user ID, content ID, and right (with parameters), as proposed in this paper. This is not a strong simplification since almost all DRM solutions (e.g., Windows DRM, Apple DRM) does not adopt any chain of licenses as described in case (i). The possibility of dynamically computing the grant of the user license on the basis of the distributor license can be used to permit the revocation of the distributor license. This capability may be a violation of the acquired rights of the final users. So that it is not commonly considered a desirable functionality.
5. **communicating grant authorization** results. The process is successfully completed when the grant authorization computed at point 4 is true, thus enabling to provide the ***Protection Information*** to the device which allows the decrypting of the content and the exploitation of the requested right. In this context, the player/device has the duty of enforcing the control of the specific rights to be exploited.

In order to reduce costs, especially as to the most expensive phases of the DRM, the core idea has been to adopt a P2P to store and retrieve the above mentioned key information regarding the ***Enabling Code*** and the ***Protection Information*** (that is the results of the grant authorization request). The P2P network model and protocol adopted have been a DHT solution since it may be managed as a distributed database and store of replica, and these replicas are located in closer nodes in terms of the hash code, that typically are not geographically closer. Please note that in [3], the P2P network has been used to index rights and metadata allowing users to search for content, but not as a distributed database for verification and grant authorization.

In the proposed solution, as to the ***Enabling Code***, the P2P network may need to store one code for each device, thus typically *O(D*U*K),* where: *D* are the devices per users, *K* are the number of replicas, and *U* the users; thus in our example some hundred millions of elements.

For the case of the ***Protection Information*** the complexity depends on *M* that is the number of DHT entries to be stored as accessible for each single right. *M* is the number of stored values. For the case of content distribution, the *number of content*number of users*number of rights per content*K,* thus billions of rights and information stored. For the case of patient health record it may be equal to the *number of patient*number of doctors per patience*number of elements per content*number of rights per element*K.* Thus billions of rights and pieces of information can be stored.

The adopted DHT P2P model allowed storing a DHT entry payload for each estimated hash key of 20 bytes, while the space for storing payload values to be stored/recovered associated with the key, can be of 1024 bytes. In particular, the data stored into the 1024 bytes have to be protected and thus are:

- a Payload Key of 256 bytes encrypted with RSA by using the public key of the PMS that can be decrypted only by the private key produced by the PMS for the specific user;
- an effective payload of 768 bytes encrypted with a symmetric algorithm by the Payload Key encrypted and contained in the first 256 bytes. Variations can be applied in this phase to increase security by combining the Payload Key in different manners and with other pieces of information.

In turn, the information stored into the 768 bytes effective payload of the P2P DHT are the:

- ***Enabling Code.*** To this end, the 20 bytes of the DHT key are estimated by using the hash code of the specific tool/device identification code; namely, the AXTID (AXMEDIS Tool ID), obtained during the first certification of each specific player and/or device.
- ***Protection Information*****:** to this end, the 20 bytes of the DHT key are estimated by using the hash code of the composed information consisting in: user ID, content ID and of a coding of right with its parameter. The Protection Information can represent the code for accessing to one protected content item or to a set of protected content items with the same model. This approach can be used to manage licenses for channel access. Thus, the "user ID" can represent a user category or domain and the "content ID" could represent a content channel, or collection. This generalization is useful for modeling domain licenses of MPEG-21 and ODRL.

The described solution allows to store and retrieve frequently used information in the DHT P2P network. Please note that, with the DRM P2P sensitive information is stored in both the final users' devices and in the producers' ones, while the security is guaranteed by the two encryption steps mechanism. This DHT P2P networks may be set to have data replica, and they are quite robust to network evolution, but at the same time it may happen that part of the data could not be accessible in some period of time. On the other hand, the DRM solution has to maintain his robustness by detecting failures and providing the grant authorization via the PMS Server in the case of missed data from the DHT P2P. Therefore, the P2P solution is an additional possibility with respect to the centralized solutions for DRM grant authorization. This means that, the information related to a new tool certification and new licenses has to be primarily stored into the central License Server (PMS) (and indirectly into the AXCS), and then into the DHT P2P network to reduce the connection workflow of the PMS. The DHT can also store the Action Log information with the same modality of the Protection Information. On the other hand, it may happen to recover the Protection Information from the P2P and to get a miss from recovering the Action Log information that has also to be updated. Then the DRM model has to decide if the access / update of this information is needed at each right exploitation or can be delayed for a while, thus reducing and relaxing the instantaneous precision of rights consumption estimation. Therefore, in the rest of the paper, the focus is kept on the Enabling Code and Protection Information when they are managed into the DHT.

To this end, Fig. 4 reports the tool certification process, which is carried out the first time a new client tool related to a registered person is added and associated to the user account. The first 7 steps are exactly the same as those performed in the centralized AXMEDIS MPEG-21 DRM, while the other steps have been added to save the ***Enabling Code*** into the DHT. In the first steps, the PMS Client (in the client device/player) performs a SOAP call (2) to the PMS Server; the PMS Server performs the certification asking to the AXCS (3) and (4) and it gets the result (5); such result is transferred to the Client device (6), (7). In order to keep aligned the DHT, the key is requested from the AXCS DB (9), (10); encrypted (11) and passed to the DHT (12), (13), (14) and (15).

Figure 5 describes the storing of a new license and the corresponding estimation of the values to be stored into the DHT. In this case, the phases from 1 to 16 are those of the former DRM, while the successive phases from 17 to the 27 deal with saving the ***Protection Information*** related to the content, user, license right into the DHT. As a first step, the PMS Client of the license production tool (namely the AXMEDIS Editor, or the License Editor, or the AXCP automated production tool [6, 8]) sends the license (1) with a SOAP call to the PMS Server. The PMS server contacts the AXCS for the verification (5), then the license is verified (6), (7); the license is evaluated to understand if it is a distribution license or a final user one (8), (9); different algorithms are used to verify and store such licenses (10), (11), (12); In the case of final user license, all the licenses data have to recovered and verified again into the AXCS (13), (14); to send the resulting license ID (15), (16) to the

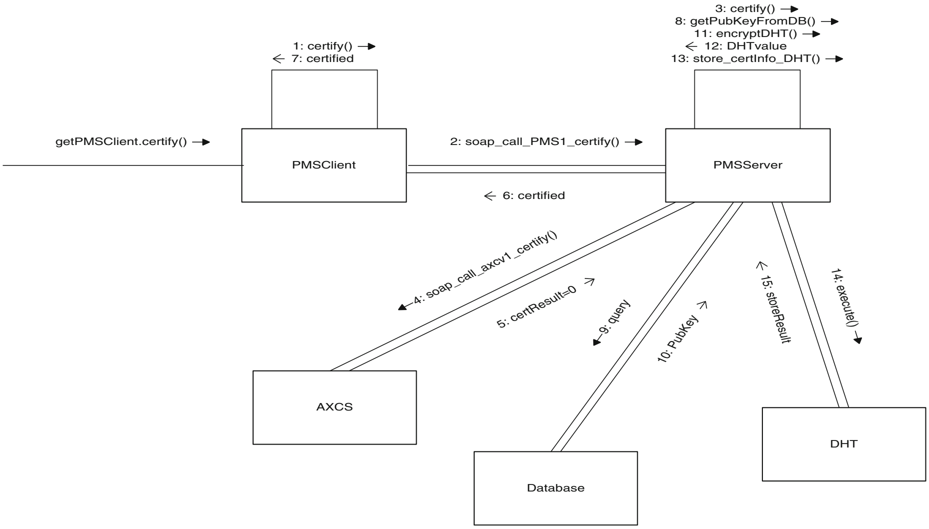

Fig. 4 Certification process of DRM P2P

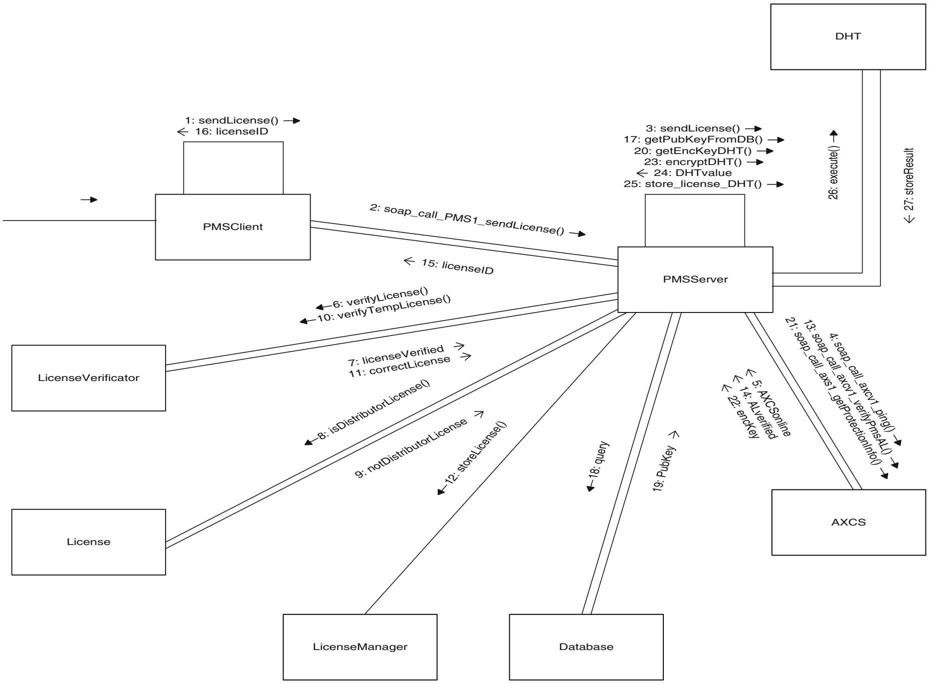

Fig. 5 Computing and saving protection information related to a grant authorization formalized in a license, in the case of DRM P2P

production client, thus confirming the operation as successful (which is to say, the license has been produced, it is valid, and in the case of final user license, it is coherent with some mother licenses). After these phases, the PMS Server autonomously starts a process of storing the information into the DHT P2P. To this end, it recovers key from the database (17), (18), (19); encrypts key (20); recovers ***Protection Information*** from the AXCS (21), (22); encrypts the value for the DHT (23), (24); and posts the value corresponding to the license (User ID, rights, content ID, etc.), namely the protection information into the DHT P2P (25), (26) and (27).

From the point of view of the client players/devices the acquisition of both the verification and the permission information are much simpler. Therefore, the enabled P2P DRM DHT AXMEDIS player has been developed by implementing a limited number of changes within the former player. The main changes consisted in adding an independent process to manage the DHT. The DRM P2P integration is grounded on the fact that the client tools (players / decoders) first try to perform the tool verification via the P2P DHT network, and once verified, the grant authorization is requested on the P2P network to get the Protection Information stored into the DHT as depicted in Fig. 5. If the verification is successfully performed from the DHT, and yet the grant authorization on the DHT fails for the lack of data, the client has to restart the process from the DRM PMS central servers. When the verification on the DHT P2P fails, the client goes directly on the PMS Server to perform the verification. Thus, the successive step of grant authorization is first tried from the DHT. In order to realize a more robust P2P network in terms of replica and data availability, every time a miss is detected the information is put again into the DHT by using sequences similar to those depicted in the Figs. 4 and 5.

The main goal of the adoption of the DRM P2P is to reduce the PMS Server workload especially in the event of many “contemporary” requests. This kind of condition could be fulfilled in a centralized manner by scaling up the PMS Service with multiple PMS Servers, a balancer, and a large network bandwidth capability. On the other hand, the adoption of a stronger PMS Server would lead to a cost increase of the DRM infrastructure and management, and thus it would limit the global scalability of the DRM.

Moreover, the P2P DHT storage can be directly activated into the client player or as service into the several devices of the final user and/or of producers (obviously informing the user during the installation to get the authorization to install the DHT as a stable service). This latter solution is more reliable and effective to implement the P2P DHT as a service running in any case, also when the client player is not running; thus increasing the node availability. The implementation of the P2P node service has been performed by customizing the Bamboo DHT library [4]. For the Bamboo DHT behavior at regime for long periods of work, please see [24, 25].

5 DRM P2P performance analysis

The main operations to be performed into the DHT P2P network are related to PUT and GET the above mentioned pieces of information (i.e., Enabling Code, and Protection Information). In addition, the removal of the same information could be used to (i) revoke licenses/content when they are black-listed, out of date or not needed any longer, and (ii) remove users when they are banned or their registration has expired. This allows keeping clean the DHT. Moreover, it could be very useful to remove licenses that have ended their time span; for example the play for two days, the play once for a Video on Demand (that typically is implemented as a grant to play that content within 24 hours or less), etc. An explicit removal/delete of the information can be very expensive or unfeasible: P2P nodes can be offline in the moment of cleaning, thus forcing to repeat the action. On the other hand, the DHT can be kept clean by imposing a *TimeToLive* (*TTL*) for each stored information. For example, for the Protection Information the license expiration time, and for the Enabling Code the user/device certificate expiration. Moreover, 1 year license can be more easily controlled if it is issued as a new license every month, for the next 12 months.

According to the DHT P2P, each node of the P2P network has only a partial view about the whole stored data in the DHT P2P. In particular, each single node knows its neighboring nodes (called leaf set) and the routing table to propagate searches. When a node enters in the network, it contacts a close node and starts creating its view of the leaf set and the routing table. According to the DHT P2P network model, the insertion/search of every elementary DHT key-information couple is performed with respect to the node which has the closest ID with respect to the key to be saved/retrieved. This means that the network allows navigating from one node to another to perform the put/get operations, according to a specified algorithm. In general, if the network structure and algorithm are known, the maximum number of hops to reach a node with a given ID from the starting node ID can be pre-estimated.

The arrival and departure by a large percentage of the active peers/nodes (the so called churn) is the most destructive activity in the DHT since the stored data may disappear [18, 24]. To cope with this problem, the DRM DHT P2P networks should have the advantage of providing a certain degree of robustness by using a number of K replicas distributed in the neighboring nodes of each node. The adoption of larger values for K increases the probability of successfully getting an information also in case of large churn. Moreover, the DHT

P2P network is intrinsically fault tolerant due to data and node redundancy and the lack of a central server, (e.g., robust with respect to the failure of nodes). It also presents a certain degree of scalability since the computational complexity for the operations of Get/Put is an O(Log E), where E is the number of peer elements in the network.

Another very relevant case for DRM P2P solution consists in a large number of DRM clients and thus of P2P nodes performing their requests at the same time. For example, to request the grant authorization to access an event or to see a media content. This may happen just after the withdrawal of a large percentage of nodes. For example, when moving from a live pay per play event to another, or at the end of the working hours in the doctor office.

In this section, the performance analysis of the above described DRM P2P solution is analyzed. The work presented in this section does not aim at demonstrating the effectiveness of Bamboo library that we have used in the development, but at assessing (i) the effectiveness of the usage of a DHT P2P solution for some DRM critical and specific cases, and (ii) the identification of reasonable values of DHT parameters to cope with critical conditions which may occur according to the typical behavior of the users involved in the DRM applications, and in particular with respect to the churn effect. The specific intention was to replicate the adopted models into a real implementation where the P2P Library has been adopted for the DRM.

Therefore, in order to validate the solution in realistic conditions, a large number of nodes and key-value couple information stored in the DHT should be needed. On the other hand, to set up a realistic network of millions of devices/users for validation purposes is not feasible, and in large measure it is neither feasible to have 1000 computers with 10000 nodes each to perform a scaled simulation. For this reason, some simulation sets have been focused and performed to recreate the conditions in terms of *data per node* in which the solution could be analyzed when large scale data are stored. Details are reported hereafter.

For the simulations, the OverSim simulator (http://www.oversim.org) has been used. It is based on OMNeT++ [23]. The advantage of using OMNeT++ consists in its capability to simulate (i) the communication stack and delays aiming at reproducing the condition of real applications, and (ii) on top of the communication stack the P2P network with different routing algorithms. OverSim allows simulating P2P networks with realistic conditions (creating profiles for describing withdrawals and arrivals of new nodes) with several different churn models. This churn evolution can be modeled as distribution trend: exponential, random, Pareto, and by imposing different parameters. Figure 6 shows the simula-

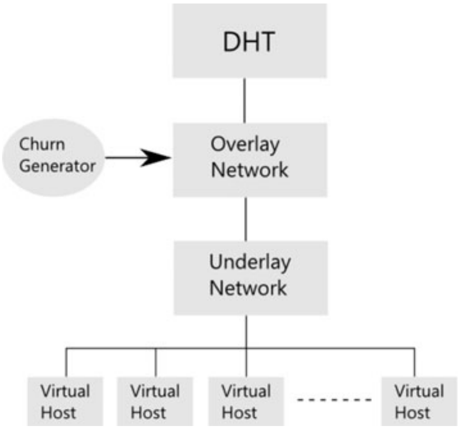

Fig. 6 Architecture for P2P DHT on Oversim

tion architecture. The Bamboo DHT network (Overlay + DHT) is simulated by OverSim, and the structure of the underlying network by the communication networks simulation framework INET. Both packages are additions for the simulation environment OMNeT++. Churn generator is a module responsible for the disconnection distribution simulation of participants in the overlay network.

The adoption of the OMNeT++ simulator allowed to simulate the network stack under the overlay network specifying many parameters, see Table 1, for the most relevant parameters as Delay, Jitter, and SendQueueLengh. We decided to adopt a coherent model in simulation with respect to the actual solutions implemented in the player and network. OverSim provides an implementation of the Bamboo model including the UDP layer. At low level, the adopted communication protocol has been the UDP, as it has been also used by Bamboo [25]. In Table 1, the adopted constant component of transmission delay was in the range of 30–50 ms; which is the offset value of the delay [5]. According to OverSim, the delay is imposed adding to the offset a delay computed on the basis of the simulated 2D network topology, which means creating a random distribution of positions in a 100 by 100 area and computing the delays on the basis of the Euclidean distance. The OverSim also adds a delay according to the packet side, while in the proposed solution the packet size is of 1 kbyte and therefore this aspect, that also simulates the costs to reassembly data blocks, has a marginal impact. As a conclusion, the typical maximum value of delay was about 165 ms and the mean about 100 ms. In the following reported cases, 50 ms has been used as offset delay.

In the simulations, the occurrence of a “missing” (failure in getting a value) from the DHT may be due to: (i) lack of requested key-value; (ii) lack of network problems due to connection, etc.; (iii) wrong key to get the value; (iv) expired *TTL* for the key (expired license and/or certificate); (v) key value not accessible (“lost”) on the DHT due to lack of connection of the nodes that may have it; (vi) too high workload of some nodes, which does not allow nodes to provide answers. Case (v) and (vi) can be reduced by tuning the DHT parameters. For example, case (v) can be reduced increasing the number of replicas, at the expenses of each single node’s memory of. Please note that the simulated solution aims at keeping constant the number of replicas when some nodes leave the network. If all nodes storing a given key-value leave the network at the same time (churn), the situation cannot be recovered; while if some of them leave the network, the network regenerates other replicas. Case (vi) also depends on replicas since with high values of replicas the value recovering can be performed from other nodes. A high workload per node may be due to a very high number of replicas or to a very low number of nodes with respect to the DHT size.

As to the considered DRM P2P simulation conditions, the setup of *TTL,* implying case (iv), is not very relevant. Critical conditions are mainly present when a large number of nodes leave the network in nearly same time bringing with them valid payloads. Therefore,

Table 1 Main parameters of the DHT P2P network in the experiments

Parameter	Description	Range of values
L	Size of the leaf node list	8–400
Ltime	Time to update the leaf node list	25–250 s
K	Number of replicas per node	4–200
N	Number of nodes involved in the network	1000–10000
Delay	Offset delay of transmission	30–50 ms
Jitter	Variability over time of the packet latency across a network	10 %
SendQueueLengh	Max dimension of the output buffer	0.5 Mbyte

during the whole simulation we assumed to have in the network relevant and useful information, disappearing only due to the churn effects.

In most of the above mentioned failure cases, the client fails in accessing the payload from the P2P, so that it starts to perform the request to the central PMS Server (Licensing Server). Typically, the lack of getting an expired payload/information from the network cannot be considered a problem for the DRM, since it implies in any case the failure of the grant authorization algorithm. On the other hand, the return of a miss from the DHT due to case (iv) would lead the client to perform the request to the central PMS Server, since the requested license or certificate have expired, but a new one could be available on the PMS Server due to a new license acquisition by the user. A client could start requesting expired information, just to put in trouble the system. On the other hand, according to the model the user and/or device insisting in performing this kind of actions (deny of service) can be banned placing user or device in black lists and thus blocking the transaction in the first verification protocol.

The adopted DHT P2P, based on Pastry, has a key of 20 byte; the space of keys is of 2^{160}, with a payload of 1Kbyte. The size of the payload is not influencing very much the network performance, but the size of the memory allocated on the single node, instead. The key space is circular as in other DHT, while the node ID is taken uniformly and not distributed on the basis of geographical information.

The routing overlay network is created by discovering other nodes to create a list of leaf nodes of *L* closest nodes (see Table 1). The time adopted by a node to perform the check and update of its leaf node list is called *Ltime.* Every *Ltime* the node contacts one of its closest nodes to exchange the leaf node lists, thus maintaining the network nodes connected, and covering the possible disconnections with new entries. For high values of *L*, the *Ltime* has to be large enough to leave time to the node to work and check the list.

5.1 Experimental results

In this section, the most interesting results related to the assessment of the DRM DHT P2P solution with respect to critical cases are reported. To report the whole results performed in the different cases and values, as described in Table 1, would be too space and time-consuming and not useful to understand the effective results of this study.

A significant simulation has been performed by loading a DHT P2P network with about 100 payload values per user, and considering a network segment of 1000 nodes, *N* (client users). Thus a total of 100.000 DHT entries / information stored. In the context of DRM, to have an average of 100 payloads per user means to have an average of 100 rights/licenses stored in the network per user. This is quite realistic for both cases: elements of the personal health record or as acquired content items in pay per view. Moreover, if the number of replica *K* is 200, each node/user client device could have a mean of 20000 elements per device. This means to have about 20 Mbytes of memory space per device, thus allowing the usage of the solution in classical media players for both PC and mobiles. These values can be applied to a large network of 100 millions of users, for each of them 100 values stored, 200 replica, etc., and the same number of elements in the client device memory. The following experiments have been performed to analyze these conditions.

Each experiment started with a setup phase to reach full capacity running conditions. The setup phase consisted in:

- establishing a DHT P2P network with *N* nodes, putting them in execution and running conditions,

- loading key-payload couples in the DHT P2P network (the information loading is carried out by a single node as in the case of the PMS Server described in Figs. 5 and 6), and
- waiting for stabilizing the replica distribution in the DHT P2P network.

After the setup, a phase of test has been performed by requesting to the network a number of keys to verify the availability of all the data in the DHT (which means to start from the condition where the missing level is zero), the stabilization of the network and measuring the latency in responding to the requests, etc.

After the setup phase, a large percentage of nodes (from the 30 % to the 75 % in different cases) has been removed from the network in less than ½ of the *Ltime*, to put in trouble the whole system. A time interval for removing the nodes greater than the *Ltime* would permit the network to reorganize and spread the list of leaf set among the closed nodes. *Ltime* was set to 250 s, for the reasons reported hereafter.

On this basis, the response of the DHT P2P network has been assessed by analyzing the response provided to 1000 requests of payloads (recovering values from the DHT entries by key) from a uniform distribution of keys and nodes, as it would happen from a uniformly distributed set of users, thus performing for each of them 1–2 requests in the different cases. The results are reported in the following figures. From the operative point of view each simulation took about 3 days on a 16 Gbytes of RAM, Intel Xeon 2.8 GHz.

Figure 7 shows the trends of the most relevant measures obtained from the DHT P2P for the most relevant cases. The latency is the time of response of the DHT P2P network, when a node asks for a key-value couple. In particular, with high values of replicas and a leaf set imposed to ½ of the number of replicas. From the figures, it is evident what follows:

- **mean number of keys** per node remained stable and identical in both cases (a) and (b).
- **percentage of missing** is quite stable with respect to the churn size, as depicted in Fig. 7 and in large scale in Fig. 8.
- rest of the other figures have been scaled up of a factor of 2, by passing from first conditions (a) to the second (b) in which both *L* and *K* have been doubled.
- **latency** (the response time of the network) is linearly dependent with respect to number of replicas.
- **P2P network** responds quite well with respect to the disappearing of a high quantity of nodes even if the nodes disappear in a very short time with respect to *Ltime*. On the other hand, the number of miss drastically increases when the percentage of leaving nodes is greater than 50 %.

The *Ltime* parameter has been set to 250 after the analysis reported in Fig. 9. From the graph a minima has been identified close to that value. The estimated trend reported in Fig. 9 is due to two competing factors. When *Ltime* is low, the node workload to update the leaf set is high since it is engaged in frequent connections with the other peers. On the other hand, increasing *L* leads to increase the probability that some segments of the network are lost when a large number of peers/nodes leave the network. In [28], it has been suggested to use *Ltime* smaller than 360 to maintain the network stable.

As a conclusion, the adoption of DHT P2P for storing the DRM information is a viable solution to reduce the connection workload of the central Licensing Server, namely the PMS in the AXMEDIS MPEG-21 case. The effective adoption of the solution allows scaling up the DRM without the need of having a strongly powerful infrastructure. The applicability conditions of the proposed solution are related to the number of licenses per user and replicas per license which is the average number of DHT entries in each P2P node – e.g., *K*licenses per user.*

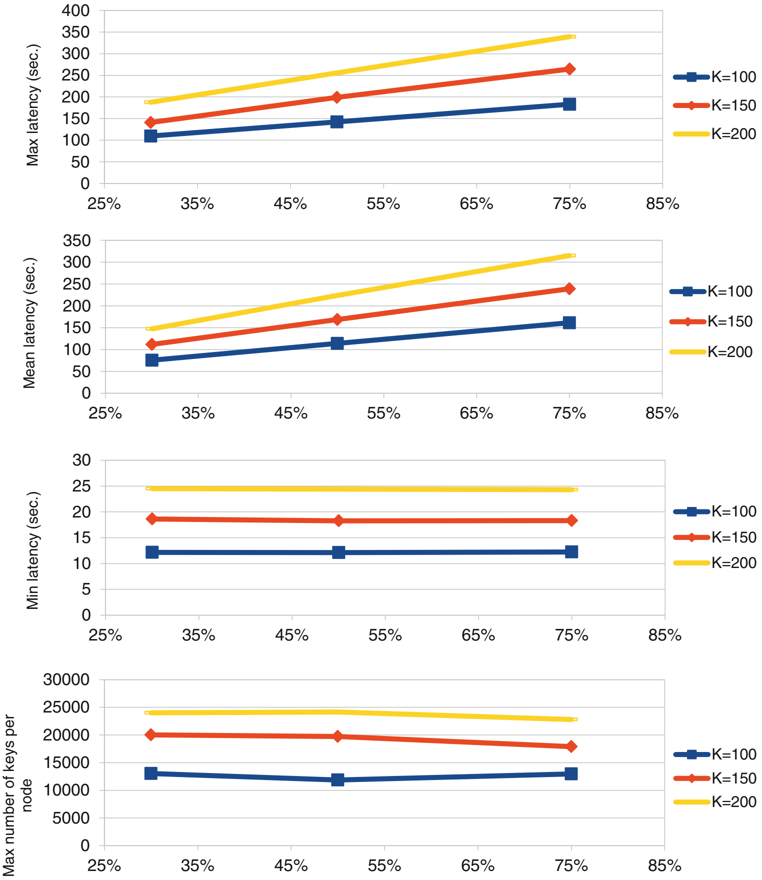

Fig. 7 Trends of the most relevant measures performed on the DHT P2P for the cases presenting a strong reduction of nodes, equal to 30 %, 50 % and 75 % of churn, with: $L=2$ K and $K=100$, $K=150$ and $K=200$. The trend of the minimum number of keys has not been reported since the value is almost constant and equal to 1000 keys per node for any value of churn and K, the changes are in the order of 3 %

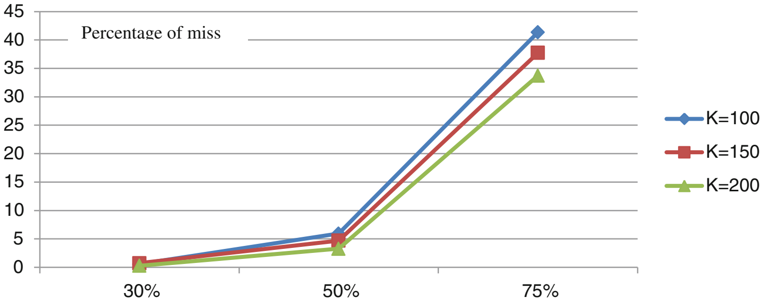

Fig. 8 Trend of the percentage of missing, estimated on the DHT P2P for the cases of a churn of 30 %, 50 % and 75 %, and K equal to 100, 150, and 200

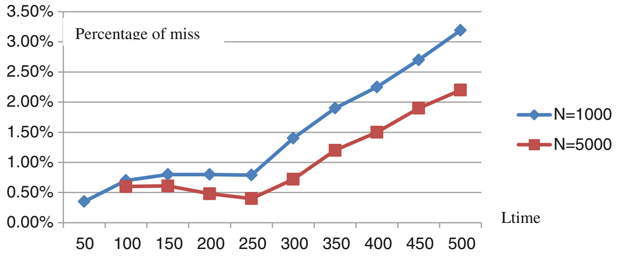

Fig. 9 Trend of the percentage of missing estimated on the DHT P2P. With respect to the Ltime value for the cases of a churn of 30 %, and different values of the number of nodes, N, with the same number of replica

For the case of TV events, it is quite probable that the average number of DHT entries per node is satisfactory since most users would be connected with their device. In this case, the number of replicas can be maintained low, thus increasing the reaction time in any node disappearing. For example, in Fig. 10, the trend of percentage of missing analysis, with 10 licenses per users, is reported. The graph reports the trend of network with different values for *K* (replicas) (where *L* was set as above at *2*K*). When *K* is bigger than the number of licenses per user, the network shows a good performance level. Also in this case, the limit of the 50 % for the churn has been detected. In fact, even in presence of 50 % of churn, more than 80 % of the requests are going to pass from the DHT, thus reducing the workload of the PMS to only 20 %. This reduction of workload implies a reduction of costs for the PMS Server that needs to have only 20 % of network bandwidth and CPU usage.

For the case of EMR, the rights to be stored are related to the whole population, while only a small percentage of them would be engaged in accessing the record at the same time; this means that only a small number of nodes/players would be active. Moreover, the same user may ask to get access to several DHT entries, since the patient record may have content provided from different sources and thus different protection information. For this reason, the closer conditions are reported in Fig. 8. In those cases, the number of replicas has to be high to increase the probability that the data entry would not disappear from the network. From the case of Fig. 8, a reduction of PMS workload higher than 50 % has been obtained even when 75 % of nodes disappeared from the network. According to Fig. 8, a churn of 75 % with *K= 100* leads to have a reduction of network costs for the PMS server which corresponds to more than 55 %. Figure 11 reports the comparison of PMS Server network bandwidth as a function of the number of requests, both considering cases with and without the P2P DHT network support. From the graph, the cost reduction is strongly evident. A similar graph can be obtained for the CPU usage, since a reduction of requests arriving at the PMS Server also implies a reduction of CPU usage. In order to be able to exploit such cost reduction, the PMS Server could be deployed in cloud servers where the CPU and network

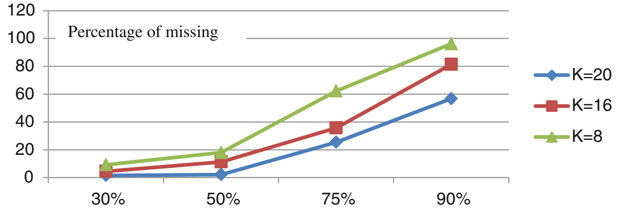

Fig. 10 Trend of the percentage of missing estimated on the DHT P2P with respect to different percentage of churn, for *K* equal to 20, 16 and 8

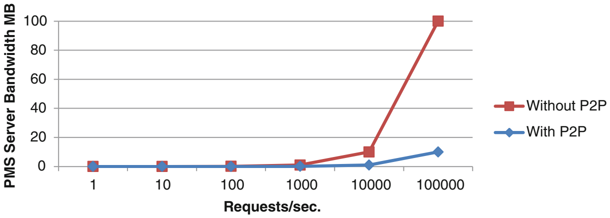

Fig. 11 Trend of needed bandwidth for the PMS Server with respect to the number of requests per second of verification/authentication coming from the clients both with and without P2P solution, for $K=100$

usage can be paid “as a service. See for example the elastic exploitation of resources according to service license agreement of public cloud solutions in Amazon EC.

Moreover, as mentioned in Section 5.1, the final user DRM clients could be no more complex than the classical DRM based solutions without P2P support. This has been verified in the cases of MPEG-21 AXMEDIS player for PC, and web browser plug in players. The increment of memory usage has been in the order of few dozens of Mbytes.

On one hand, the percentage of users with active P2P nodes (accessing or not their own EMR) could be smaller than 5 % of the population for which the DHT entries have to be provided. On the other hand, a number of client tools would be quite stable by taking into account those located in the hospitals, from the doctors, etc. We have in Europe about 280 people per doctor, this means that we can suppose to have a stable P2P node every 500 users (thus 80000 nodes for 80 millions of users in a nation: German and Italy fall within that range), and 80 billion of DHT entries (with 10 licenses per user, and 100 replicas). In these conditions, a different solution has to be used, for example by adopting a set of super peers as in [7], where the super peers are storing content files in their protected encrypted file format by using controlled algorithms. According to this worst case, it means to have 500.000 DHT entries per client (that is 500 Mbytes of HD space used for the DHT). This kind of peer node client, different from the peer node supposed to be used by the regular users (as described in Section 5.1), may be engaged in providing response from 1 % of its DHT entries. So that, it may receive about 5000 requests of 1Kbytes each. These numbers can be supported by any client device connected to some DSL devices. Moreover, if these super peers are supposed to be maintained active, the number of replicas can be reduced to 10, thus reducing the size of their memory to 50Mbytes which is again a memory size comparable to that of regular clients mentioned in Section 5.1.

6 Conclusions

Digital rights management, DRM, solutions are today quite widespread in industrial applications. The complexity and costs for granting authorization according to some licenses is typically demanded to a single central service which has to provide: (i) suitable storage for the information, (ii) computational capabilities to compute the logical rules, and (iii) networking capabilities to cope with millions of users requesting the grant at almost the same time. The focus of this paper has been on reducing DRM costs by solving the scalability problems behind the complexity of granting authorizations and on performing verification for a large number of users, content and rights associated with them. The studies and the solutions reported in this paper have been worked out and validated on top of MPEG-21/AXMEDIS DRM solutions and tools. The proposed solution is based on the exploitation of the DHT P2P network and storage to cope with verification and grant authorization. The main difficulties addressed in the paper have been (i) to find a solution to store DRM information to perform in a P2P manner the verification and licensing, including the information shape to be stored (see Section 4), while maintaining the security level and tuning the P2P parameters, (ii) to restructure both client and server, while maintaining the compatibility with the client–server solution without P2P, (iii) to prove that the solution can be suitable and may effectively reduce the costs despite the user behavior in terms of churns. The paper reported the AXMEDIS DRM solution and services in terms of service graphs and action diagrams. This allowed highlighting the points where DRM solutions can be adapted to exploit the P2P capabilities to reduce the complexity and costs of the DRM service, by considering scenarios related to the user verification and grant authorization from licenses. The identified parameters have shown that it is possible to obtain a strong reduction of costs even in the present of churn greater than the 50 %. Moreover, as mentioned in Section 5.1, final user clients could be no much more complex than the classical DRM based solutions without P2P support.

The solution proposed is general enough to be adopted in other DRM solutions since the information stored into the P2P refers to the grant authorization and to the certification of device which are functional to almost all DRM solutions. The performance analysis of the DRM P2P solution aimed at fitting the DHT P2P into the specific cases by stressing the most relevant parameters.

The experimental results highlight the parameters' values of the DHT P2P which are suitable for the applicative scenarios: the distribution of media content, and the management of security for EMR/PHR. The work presented in this paper does not pretend to be an exhaustive analysis of the security and protection problems of EMR, but only stresses that DRM P2P can be one of the needed technologies to reduce DRM complexity.

Acknowledgments The authors would like to express their thanks to AXMEDIS partners, to the Expert-User-Group and all affiliated members for their contribution, support and collaboration; to the European Commission for the co-funding, and to Paolo De Luca who specifically helped us in the development of some of the P2P services.

References

1. Allasia W, Chiariglione F, Difino A, Gallo F, Milanesio M, Schifanella R (2008a) "Digital Rights Metadata Management and Retrieval on Structured Overlay Networks", Proc. of the 9th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'08), May 7–9, Klagenfurt, Austria, pp. 130–133, IEEE Computer Society, 2008. (http://doi.ieeecomputersociety.org/10.1109/WIAMIS.2008.33)
2. Allasia W, Chiariglione F, Falchi F, Gallo F (2007) An Innovative Approach for Indexing and Searching Digital Rights, Proceedings of the 3rd International Conference on Automated Production of Cross Media Content for Multi-channel Distribution (AXMEDIS'07), November 28–30, Barcelona, Spain, 2007, pp. 147–154, ISBN 0-7695-3030-3, IEEE Computer Society, 2007 (http://dx.doi.org/10.1109/AXMEDIS.2007.4)
3. Allasia W, Gallo F, Milanesio M, Schifanella R (2008b) "Governed Content Distribution on DHT Based Networks", ICIW '08. Third International Conference on Internet and Web Applications and Services, Page(s): 391 – 396, 2008, doi:10.1109/ICIW.2008.40
4. Bamboo DHT, http://www.bamboo-dht.org

5. Baumgart I, Heep B, Krause S (2007) “OverSim: A Flexible Overlay Network Simulation Framework”. Proc IEEE Global Internet Symp 2007, pp.79 – 84
6. Bellini P, Bruno I, Cenni D, Nesi P (2012) “Micro grids for scalable media computing and intelligence on distributed scenarious”, IEEE Multimedia, Feb 2012, Vol.19, N.2, pp.69.79, IEEE Computer Soc. Press
7. Bellini P, Bruno I, Cenni D, Nesi P, Rogai D (2007a) “P2P Architecture for Automated B2B Cross Media Content Distribution”, Automated Production of Cross Media Content for Multi-Channel Distribution, 2007. AXMEDIS '07. Third International Conference on, AXMEDIS 2007, IEEE press, 28–30 Nov. 2007 Page(s):105–112, Digital Object Identifier 10.1109/AXMEDIS.2007.31
8. Bellini P, Bruno I, Nesi P (2006a) “A language and architecture for automating multimedia content production on grid”, Proc. of the IEEE International Conference on Multimedia & Expo (ICME 2006), IEEE Press, Toronto, Canada, 9–12 July, 2006
9. Bellini P, Bruno I, Nesi P (2011) “Exploiting Intelligent Content Via AXMEDIS/MPEG-21 For Modeling and Distributing News”, International Journal of Software Engineering and Knowledge Engineering, World Scientific Publishing Company, Vol.21, n.1, pp.3–32, 2011
10. Bellini P, Bruno I, Nesi P, Rogai D (2007b) “Architectural solution for interoperable content and DRM on multichannel distribution”, Proc. of the International Conference on Distributed Multimedia Systems, DMS 2007, September 6–8, 2007, San Francisco Bay, USA, Organised by Knowledge Systems Institute
11. Bellini P, Nesi P (2005) “An architecture of Automating Production of Cross Media Content for Multi-channel Distribution”, Proc. of the first International Conference on Automated Production of Cross Media Content for Multi-channel Distribution, AXMEDIS 2005, 30 November – 2 December 2005, Florence, Italy, IEEE Computer Society press
12. Bellini P, Nesi P, Ortimini L, Rogai D, Vallotti A (2006b) “Model and usage of a core module for AXMEDIS/MPEG21 content manipulation tools”, Proc. of the IEEE International Conference on Multimedia & Expo, ICME 2006, IEEE Press, Canada, 9–12 July, 2006
13. Bellini P, Nesi P, Rogai D (2007c) “'Exploiting MPEG-21 File Format for cross media content”, Proc. of the International Conference on Distributed Multimedia Systems, DMS 2007, September 6–8, 2007, San Francisco Bay, USA, Org by Knowledge Systems Institute
14. Carrion Senor I, Aleman JLF, Toval A (2012) “Personal Health Records: New Means to Safely handle Health Data?”, IEEE Computer, November 2012, pp.27–33
15. Iannella R (2002) “Open Digital Rights Language (ODRL)”, Version 1.1 W3C Note, 19 September 2002, http://www.w3.org/TR/odrl
16. Kuhlisch Kraufmann B, Restel H (2012) Electronic Case Records in a Box: Integrating Patient Data in Healthcare Networks”, IEEE Computer, November 2012, pp.34–40
17. Lin ET, Eskicioglu AM, Lagendijk RL, Delp EJ (2005) “Advances in Digital Video Content Protection”, Proceedings of the IEEE, Vol.93, N.1, pp.171-183, January 2005
18. Liu D, Feng C, Gang Y, HuaiMin W, Peng Z (2010) “LSB-Chord: Load balancing in DHT based P2P systems under churn”, 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), Page(s): 466 – 470, 2010
19. Mourad M, Hnaley GL, Sperling BB, Gunther J (2005) “Toward an Electronic Marketplace for Higher Education”. Comput IEEE pp.58–67
20. MPEG Group MPEG-21 DID, “Introducing MPEG-21 DID, Digital Item Declaration”, www.chiariglione.org/mpeg/technologies/mp21-did/
21. MPEG-21, http://mpeg.chiariglione.org/standards/mpeg-21/mpeg-21.htm
22. OMA DRM 2.1: http://www.openmobilealliance.org/technical/release_program/drm_v2_1.aspx
23. OMNeT++: http://www.omnetpp.org/
24. Rhea S, Dennis G, Timothy Roscoe, and John Kubiatowicz (2004) “Handling Churn in a DHT”, Proceedings of the USENIX Annual Technical Conference, June 2004
25. Rhea S, Godfrey B, Karp B, Kubiatowicz J, Ratnasamy S, Shenker S, Stoica I, and Yu H (2005)“OpenDHT: A Public DHT Service and Its Uses”, SIGCOMM’05, August 21–26, 2005
26. Rodríguez E, Gallego I, Delgado J (2007) “Use of MPEG-21 for License Protection and Key Management in DRM Systems”, Third Intern. Conference on Automated Production of Cross Media Content for Multi-channel Distribution, pag 163–160, 2007
27. Wang X, De Martini T, Wragg B, Paramasivam M, Barlas C (2005) The MPEG-21 rights expression language and rights data dictionary. IEEE Trans Multimed 7(N.3):408–417
28. Zangrilli M, Bryan D (2007) “A Bamboo-based DHT for Resource Lookup in P2PSIP, draft from http://www.p2psip.org/drafts/draft-zangrilli-p2psip-dsip-dhtbamboo-00.html

Pierfrancesco Bellini is a contract Professor at he University of Florence, Department of Systems and Informatics. His research interests include object-oriented technology, real-time systems,formal languages, computer music. Bellini received a PhD in electronic and informatics engineering from the University of Florence, and has worked on projects funded by the European Commission such as: AXMEDIS, MOODS, WEDELMUSIC, IMUTUS. MUSICNETWORK, VARIAZONI and many others. He has been co-editor of MPEG SMR.

Paolo Nesi is a professor at the University of Florence, Department of Systems and Informatics, chief of the Distributed Systems and Internet Technology lab and research group, and vice-director of the department. His research interests include massive parallel and distributed systems, content protection, DRM, P2P, physical models semantic computing, object-oriented, real-time systems, formal languages, and computer music. He has been the general Chair of DMS, IEEE ICSM, IEEE ICECCS, WEDELMUSIC, AXMEDIS international conferences and program chair of several others. He has been the coordinator of several R&D multipartner international R&D projects of the European Commission such as ECLAP, AXMEDIS, MOODS, WEDELMUSIC, MUSICNETWORK and he has been involved in many other projects. He has been co-editor of MPEG SMR.

Fabio Pazzaglia is a research contractor at the University of Florence. His research interests include P2P, grid computing, image processing.