International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 5 (2013) pp. 011-029

Reliable P2P Content Delivery for Alternative Business Models

H. Castro[1], A. P. Alves[1] and M. T. Andrade[1]

[1] INESC TEC, Faculdade de Engenharia da Universidade do Porto,
Campus da FEUP, R. Dr. Roberto Frias, 378 4200 - 465 Porto, Portugal
hcastro@inescporto.pt, palves@inescporto.pt, maria.andrade@inescporto.pt

***Abstract*: The production, distribution and consumption of information goods have evolved through continuous disruptions, such as the invention of the printing press. Presently, these activities are being disrupted by the Internet and digital technology revolutions, which are triggering a paradigm change in this field. P2P content delivery is included in the disrupting factors. It has endured an explosive development in the non-commercial sector. Commercial content delivering entities reacted slowly to these changes, but have also begun to employ P2P technology. This employment, however, has been hesitant and its results unremarkable. Also, commercial initiatives which employ this technology have frequently been economically unsuccessful.**

In this paper, we argue that these two realities are interconnected. We present and analyze the current P2P content delivery scenario (commercial and otherwise). Based on previously developed work, we expose the cause, of P2P's lack of success in the commercial sector, as techno-economical, rather than just technical, as it is intimately connected to the enforced business models. Finally we present our solution for a reliable commercial P2P content delivery, and embody it in the definition of an exemplifying P2P system (under implementation), that provides the necessary support for business models, adequate for on-line operation.

***Keywords*:** P2P Content Delivery, Distributed Network, Paradigm Change, Business Model, Lateral Gain Extraction, DRM, Internet**.**

I. Introduction

Throughout time, the production, distribution and consumption of information goods (IGs), and their associated socio-economic systems, have gone through numerous changes. The greatest such changes, with an impact greater event that that of the printing press, are the ones resulting from the appearance of the Internet. This innovation, and associated technologies, is shaking the centuries-old edifice of media production, distribution and consumption. In economical terms, their impact derives from the costless data reproduction, nearly costless data exchange and disintermediation of the consumer-creator relationship that they enable [4].

P2P computer interaction follows in this logical footstep, as it optimally takes advantage of the offered cost reduction potentials, and vastly disintermediates the consumer-artist and inter-consumer interactions. Naturally, the use of this technology has seen an explosive growth, which, given its low start-up costs, occurred primarily in the non-commercial sector. P2P systems operating in this sector have typically neglected security related aspects presenting a generally low reliability in that field. This has facilitated their frequent involvement in copyright infringing activities.

Established commercial actors have had trouble responding to this process. After considerable initial resistance, (which has not yet been entirely abandoned), content distributors (CDists) have begun to embrace the Internet medium for IG delivery. More recently, P2P content delivery has also commenced being commercially employed. This embrace, though, has been timid and conservative, as CDists have generally, only, transposed "Brick-and-Mortar" operating modes onto their on-line operation. This direct reutilization of legacy Business Models (BMs), has typically lead to considerable restraints on user's freedom to manipulate content, in comparison to what has become customary on the on-line world. The enforcement of such BMs and associated content manipulation restrictions has placed significant technical demands, on the content securing provisions associated to the P2P delivery systems, and the earlier have frequently failed.

The above described scenario as resulted in an overall situation (illustrative examples will be presented in the following section), which may be predominantly characterized by: frequent user rejection of the imposed content usage restrictions; user breaking of content securing measures, evasion of usage restrictions and circumvention of the original content providers; consequent failure of content governance provisions and their reoccurring patching and growing complexity; abandonment of P2P distribution; user abandonment of commercial content delivering initiatives; and economical failure of such initiates.

A new approach is thus necessary so that commercial entities may employ a P2P delivery of media content in a reliable way.

In this work, we present, in section II, the current sate of P2P content delivery employment, in the commercial and

non-commercial sectors, focusing on the comprised content delivery schemes, the associated security measures and the employed BMs. In section III, we analyze that scenario identifying the root causes for the lack of success of commercial P2P content delivery. We then infer a set of techno-economic guidelines to be employed for on-line media delivery success (section IV) and embody them in the definition of a P2P content delivery system (section V). In section VI we highlight our proposal's advantages in comparison to the state-of-the-art and, finally, we present our conclusions (sectionVII).

II. P2P Content Delivery Scenario

A. Non-Commercial Sector

Since the rise of Napster, (the first notorious P2P network), this technology has endured a prolific evolution. Numerous P2P protocols and systems coexist in today's Internet [1]. In sub-section 1, a few illustrative P2P systems will be approached. Sub-section 2 presents an overview of the field.

1) Study Cases

a) Gnutella

Gnutella began as a fully decentralized [1] protocol for distributed search on a flat topology of peers [22]. It operates as an unstructured network (no algorithm mapping specific contents to specific storing peers), thus, in order to locate a specific data item, peers must query their neighbors.

The initial content discovery method in Gnutella was flooding. This mechanism was highly resilient to peer population transience. Still it was not scalable and overburdened the network. Gnutella has thus evolved into a partially centralized system, which employs an overlay network. The system's nodes are either leaf nodes or higher level nodes, called ultra-peers. Ultra-peers are high capacity nodes which behave as proxies for the network's leaf nodes. Content discovery is now performed in an optimized manner employing the Ultra-peer sub-structure as depicted in Figure 1. This protocol includes no relevant security measures [29].

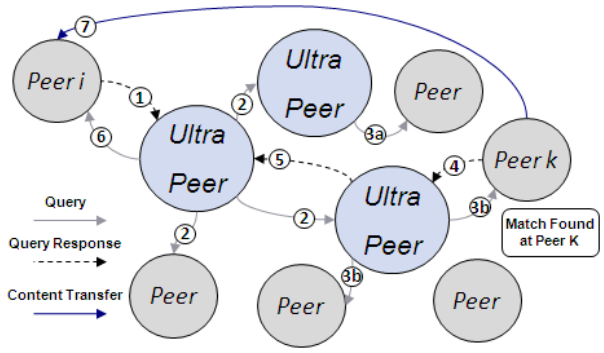

Figure 1. Gnutella Structure and Content Search

b) BitTorrent

BitTorrent is a centralized P2P protocol [23]. It employs central nodes called "trackers" for the coordination of information transfers. Trackers maintain an index of ongoing torrents (individual content exchanges) and of the involved nodes. For a client to obtain some specific content it must be given the corresponding .torrent file (which may be obtained, for instance, from a dedicated website). The client obtains, from the .torrent file, the URL(s) of the tracker(s) which is responsible for the coordination of the exchange of the desired content (the content's torrent). It contacts the tracker(s) and obtains a list of peers which are at that time participating in the exchange of the content. The client then proceeds to connect to those nodes to retrieve the content by simultaneously downloading different fractions of it, from multiple sources. The content retrieval process just described is depicted in Figure 2.

This protocol provides the following security functionalities:

- limited content integrity validation – after each content fraction is downloaded, the client checks its integrity by calculating its SHA1checksum and validating it against the corresponding valued specified in the .torrent file (which contains the sizes and checksums of all fragments). This scheme assumes the integrity of the .torrent file.
- enforcement of fair use – employs a strict reciprocity strategy for content diffusion which constitutes as a trade-based incentive mechanism, (or bartering scheme), designed to discourage free-riding. Peers upload to those peers who have uploaded to them and they download from those which have downloaded from them.

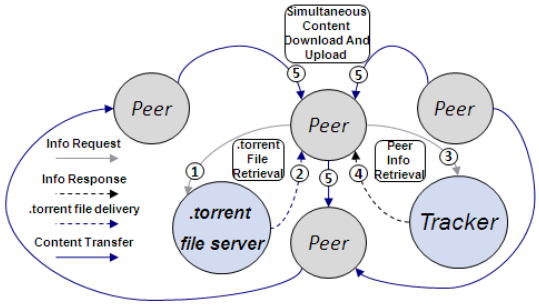

Figure 2. BitTorrent Protocol Operation

c) eDonkey

eDonkey is a hybrid decentralized P2P system. This network is composed by two functionally different types of peers – client peers and server peers. Server peers are highly reliable central nodes which are loosely connected and are run by users and the "community". They perform the role of communication hubs and content indexes, allowing users to locate files within the network [24].

At connection time, client peers register the content files that they possess, with a server peer, by providing the meta-data describing said files.

Content files within the eDonkey system are divided into blocks. A checksum is computed for each of the blocks and they are propagated between peers on demand. A checksum of all the checksums is employed by the system as the file identifier. Content may be located either by querying the server peers with values that will be matched to the descriptive meta-data that the server peers host, or by requesting a particular file through its unique network identifier. Server peers deliver, to client peers, the locations where the desired files may be obtained. Content files are then directly exchanged between client peers.

This protocol provides the following security functionalities:

- limited content integrity validation – at content retrieval time, if a downloaded block presents a different checksum value from the one specified by the peers community, the block is discharged and retransmitted [24]. This scheme assumes the integrity of the communication links through which the "correct" value of the checksum was obtained and the honesty of the involved peer(s).
- enforcement of fair use – some provisions exist but they vary with the implementation of the protocol (peer).

d) FastTrack

FastTrack [25] is a proprietary semi-centralized P2P protocol. Its constituting peers are either ordinary nodes (ONs) or supernodes (SNs). SNs occupy a more central position in the network and handle a greater and more relevant set of tasks. They perform the role of temporary content index servers. The indexed information, (delivered by the ONs, about the content that the latter are making available), consists of: the content file name: the file size; the file hash; and a set of descriptors describing the content.

Any of the FastTrack's nodes, with sufficient CPU capability and network connectivity, can voluntarily become an SN. ONs connect to a specific SN through a semi-permanent TCP connection.

FastTrack is unstructured protocol. Content discovery is performed through the employment of query diffusion over the overlay network of SNs. For each match at an SN database, the respective SN returns the IP address, server port number, and metadata corresponding to the match, to the initial querying SN which sends it to the inquiring ON.

This protocol provides the following security functionalities:

- limited content integrity validation – after retrieving a content file, the ON calculates its checksum and validates it against the value specified by its coordinating SN. This scheme assumes the integrity of the communication link between the SN and the ON.
- enforcement of fair use – the Kazaa variant of the FastTrack system, implements an incentives mechanism to reward fair use of the network's resources. The mechanism is based on the calculation, at each peer, of its level of contribution to the community. As this value is calculated and stored locally, the incentives system defined by FastTrack is obviously of little or no robustness.

2) Overview

The study cases highlighted in section II.A.1, reveal that the field of non-commercial P2P content delivery is a very diverse one. Many different types of P2P protocols exist. These present varying levels of decentralization, with Gnutella as one of the most decentralized, but also as one of the less efficient. In time a return as occurred to more centrally coordinated modes of operation, (BitTorrent, FastTrack), in order to increase operational efficiency.

In what pertains to security, these systems present very little such provisions. Those which they do offer, (typically content integrity validation and fair use enforcement), are not robust.

A number of proposals exist, mainly in the academic field, for enabling these systems to support such functionalities as anonymity provision, peer registration, content authenticity or content availability. These are generally based on distributed algorithms, typically, independent from any fixed central entity. The most robust of such proposals employ some form of collective/mutual authentication between peers. Some of these are: threshold cryptography [27], (which is not purely distributed as it requires a trusted third party), PGP [26] (which requires human evaluation of trustworthiness), or "trusted peer group" based authentication [28].

All of these solutions, however robust, still present considerable weaknesses in the face of sufficiently vast and sophisticated attacks, and lead to a considerable operational overhead. Furthermore, their adoption by the predominant P2P content delivery structures on the Internet is marginal.

Provisions expressively designed for the enforcing of content usage rights and the protection of copyright (DRM provisions), are practically universally absent from these systems.

B. Commercial Sector

1) Study Cases

Several initiatives have sprung up in the field of P2P based commercial content delivery. As it is not possible, or relevant, to address all of them, only a few illustrative cases are described below.

a) Veoh

Veoh [12], was an Internet TV service run by a California based company. It employed P2P technology (among other means), for the diffusion of commercial (and user generated) content. In regards to security provision, Veoh distributed both DRM protected and unprotected content. That is, it employed a parallel DRM structure, to enable user and peer authentication, content integrity and authenticity validation, and access control on some of its content.

It searched for years for a successful BM, but ultimately, and under legal pressure from Universal Music Group [14], declared bankruptcy in 2010.

b) Babelgum

Babelgum [2] is a free Internet TV service. Originally it employed a proprietary P2P streaming technology, but has eventually dropped it in favor of a client-server operation. The platform employs DRM protective measures which include the encryption of the exchanged content data streams [16].

Babelgum is a privately backed project which focuses on professionally produced content. It enforces some content access restrictions based on user geographical location.

Babelgum's BM is advertisement based, employing advertising revenue-sharing. Content owners receive a portion of advertising revenues. If no advertisement is associated to the content this system guarantees producers a minimum of $5 for each 1000 unique views [15].

c) Joost

Joost [6] is an Internet based system for the distribution of TV content (and other forms of video), developed by the founders of Skype and Kazaa. It employs a P2P strategy (or did so initially), for the delivery of content, under proprietary DRM protection.

Joost's employs an ad-supported BM, in a similar manner to that of regular TV, exposing users to both injected video-advertisements as well as additional interactive advertisements via overlays and short clickable pop-ups [17]. It was eventually faced with economical difficulties and remade itself as a "cost-effective" white-label video provider [18].

d) PPLive

PPLive [10] is a P2P Internet video streaming service offering un-protected content under an add-supported BM.

e) ReelTime

ReelTime was a video-on-demand provider that delivered movies and television shows over the internet. Content was delivered through a proprietary software system, (Intelligent Rapid Delivery System), which employed some P2P networking to reduce the bandwidth demands on its servers. To this end, while the most part of the content files were delivered to the users by the system's servers, a fraction of such files were transferred between the system's terminal machines (i.e., peers). The delivered content was DRM protected.

ReelTime employed a subscription and pay-per-view BM. It has subsequently ceased to operate.

f) Qtrax

Qtrax [11], supplies a legal P2P music delivery service built upon the Gnutella network. It employs Microsoft's Janus DRM technology for content protection and access control and to enforce advertisement consumption.

In time Qtrax has moved to a strictly advertisement based BM from a previous two tiered BM, which also employed subscription revenue capture.

Qtrax faced legal and financial problems that have forced it to restart its operation, after a pause period [19].

g) iMesh

iMesh [5] is a media content delivery system and an online social network which employs a Gnutella based centralized P2P strategy for content distribution. For content protection the system uses Microsoft Digital Rights Management technology [21].

Its BM allows free access to some content but relies on the permanent purchase or on the paid subscription to other content [20].

2) Overview

a) Technical Operation

Commercial, P2P based, media delivering initiatives, as shown in section II.B, generally employ either proprietary P2P solutions or pre-existing ones, which are predominantly based on Gnutella.

The details of the proprietary solutions are typically not disclosed. Still, from the few which are, it may be concluded that such solutions employ some form of partially centralized operation, often times similar to the bitTorrent protocol.

Gnutella is a purely distributed P2P protocol which is very robust, in terms of fault tolerance, but also considerably inefficient in terms of content discovery and distribution.

In terms of content access control, most of these initiatives employ classical DRM technology, that is, intellectual rights protection technology, which focuses mostly, if not exclusively, in a restrictive governance of content usages. In P2P terminology, this DRM assistance, in security matters, to the P2P delivery structure, may be described as a form of Trusted Third Party (TTP) solution. The typical resulting P2P+DRM structure may be depicted, albeit in a simplified manner, by Figure 3, (taking into consideration the notation defined bellow).

u_i = a specific system user
$psswd_{u_i}$ = password of u_i
$usname_{u_i}$ = the username of u_i

c_i = a specific system client
K_{c_i} = the public key of c_i
$K_{c_i}^{-1}$ = the private key of c_i

o_i = a specific media object
$L_{u_i}^{o_i}$ = the license governing the use of o_i by u_i

K_S = a secret symmetric encryption key
$K_S^{session}$ = a comm session encryption key
$K_S^{o_A}$ = a o_A's encryption/decryption key

K_{LS}^{-1} = the private key of the License Server
K_{UAS}^{-1} = the private key the User Authentication Server

$\mathrm{h}(x)$ = cryptographic hash function applied to x
$\mathrm{enc}_{K_{x_i}}(x)$ = the encryption of x with K_{x_i}
$\mathrm{signed}_K(x) = \mathrm{enc}_K(h(x)) \| x$ = signed x with key K
$\mathrm{secmsg}_K(x) = \mathrm{enc}_K(x)$

Its operation is as follows: At an initial moment (not displayed in the diagram), all clients will authenticate with

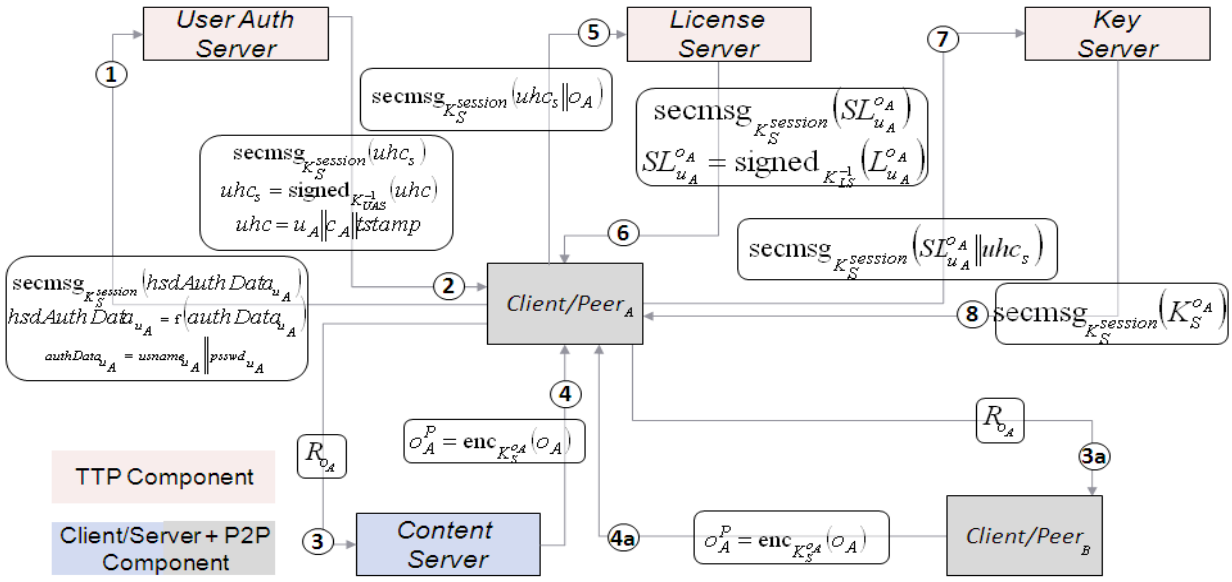

Figure 3. Typical Commercial P2P Content Delivery System Architecture

the TTP components agreeing on a common secret key ($K_S^{session}$), which will be used for a client/TTP interaction session. From that point on, all communication between a client and any TTP component will be encrypted with the agreed upon secret key (as represented by $\text{secmsg}_{K_S^{session}}(x)$). When the session terminates, the key is discarded.

Before a user (e.g. u_A) can be attended/hosted by a specific client/peer (e.g. c_A), he must first be authenticated. To do so, the user supplies his username and password, and c_A packages it into $hsdAuthData_{u_A}$, by performing some hash function on that information (employment of digest access authentication). The latter information package is sent (arrow 1) to the User Authentication Server (*UAuthS*). After validating the user's credentials, *UAuthS* sends c_A (arrow 2), a user hosting certificate (uhc_s), signed with *UAuthS* private key (K_{UAS}^{-1}). uhc_s proves that c_A is hosting u_A. If a user (e.g. u_A) whishes to consume some media object (e.g. o_A), c_A may retrieve it from the Content Server (client/server operation, arrow 3), or from another client/peer , such as c_B (P2P operation, arrow 3a). The content is obtained in its protected form, (o_A^P), which is encrypted with a secret symmetric key $K_S^{o_A}$. The client that contacts the License Server (*LS*), to inquire it about u_a's usage rights over o_A, by sending it (arrow 5), uhc_s, (proving that c_A is indeed hosting u_A), and the identification of the media object in question. The *LS* then responds (arrow 6), with the signed license ($SL_{u_A}^{o_A}$) that specifies u_A's rights over o_A (assuming u_A previously purchased such a license). $SL_{u_A}^{o_A}$ is signed by the *LS* so that its validity may be checked. c_A then sends $SL_{u_A}^{o_A}$ (arrow 7) to the Key Server (*KS*). *KS* assesses the validity of $SL_{u_A}^{o_A}$ and the rights that it grants to u_A. If all is ok, it returns (arrow 8), o_A's decryption key ($K_S^{o_A}$). c_A is then ready to access and render o_A for u_A's consumption.

In spite of their complexity and sophistication, the security services provided by the TTPs, have frequently been broken and circumvented by users [30], as this is "incented" by the uncomfortable content usage conditions that result from the operation of such security measures.

In the studied cases, the P2P delivery mode generally performs a parallel, or even just, auxiliary role to client/server delivery mode. Furthermore, the earlier mode frequently ends up being entirely abandoned.

There is also a visible decoupling of the content delivery structure from the security structure, where the later performs the mentioned TTP role, and the potential synergies between the TTP structure and the P2P content delivery structure are left unexploited.

b) Economical Operation

Commercial P2P based content delivery initiatives have been trying a variety of BMs, such as subscription, purchase or advertisement based ones. They have frequently jumped between several BMs in their search for a suitable one. There is thus no globally predominant BM in this area and the attained economical results are also unremarkable.

III. Scenario Analysis

The employment and massive popularization of P2P content delivery technologies, started in the non-commercial sector. The main concern driving the development such initiatives was the building of distributed structures capable of enabling a "good-enough" content discovery and an efficient content exchange between peripheral equipments. Guarantying content authenticity or copyright protection, were never the goals.

These technologies have had an explosive success, often in connection to copyright infringing activities. This has upset the content industry which has frequently targeted these initiatives, both technically (poisoning [7] of P2P networks), and legally. Their proliferation was unabated, though [9], [3].

Perhaps in reaction to this failure, the content industry has begun, even if reluctantly, to use P2P technology and the Internet as a content delivery platform in more innovative ways. Still, in spite of the success of non-commercial P2P content delivery, its commercial counterpart has been unremarkable, as demonstrated by the frequent cases of economical failure (e.g. Veoh or ReelTime), or P2P abandonment in the initiatives exposed in II.B.1 (e.g. Joost or Babelgum).

This panorama of failure and technological retreat has been caused, as argued in [4], by the non-observation, on the part of CDists, that the Internet medium has come to radically alter the technical, economic and social premises underlining IGs' production, distribution and consumption. This new medium virtually eliminates storage, reproduction and distribution costs, and enables the disintermediation of the artist-consumer relationship. Its establishment represents the rise of a new technical paradigm in the field of information exchange and manipulation [4].

In the "Brick-and-Mortar era", CDists controlled the content distribution structure (stores, CDs, DVDs, etc). This structure was both socially useful and inescapable for consumer access to IGs. It was the scarcity and controllability, inherent to the materiality of the distribution structure that secured a useful and profitable role for CDists. The Internet, given the automaticity and immateriality that are associated to it, is a far more cost effective alternative and eliminates the need for such a material structure. This is rendering the CDists' traditional role, useless. To regain that usefulness, CDists are adopting new platforms for their activity, apparently moving in a progressive way. Still they do so while attempting to preserve their age-old reliable BMs, thus, with regressive objectives.

This conservativeness means that many of these initiatives have employed BMs which are based on the direct the sale of media goods (or close to that), and thus, on the restrictive control of access to them [4]. That is, BMs based on the exploitation of content scarcity in a medium, (the Internet), which is adverse to the preservation of such scarcity. To secure such BMs, DRM technologies are employed in an attempt to artificially maintain the necessary IGs scarcity. The enforced access restricting practices result in uncomfortable content usage conditions that drive users away.

Furthermore, these initiatives have generally regarded P2P operation as a merely auxiliary aspect of their overall operation, and have frequently delegated the handling of such aspects (P2P content exchange), to already existing P2P structures (e.g. Gnutella), which are outside of their control and present unreliable performance. The inevitable loss of control over content, that all this implies, is in contradiction with the control levels required by the chosen BMs, and thus, P2P content exchange is frequently abandoned.

The employment of P2P content distribution technologies in the commercial sector, (event if gaining momentum), is still in an immature phase. This is so because of the anachronic attitude with which CDists have employed this technology. The maturing of P2P's commercial employment as well as the reacquisition of a useful social role, by CDists, requires that the later accept and embrace the changes brought by the Internet revolution, relinquish absolute control over their information goods, and employ radical new BMs which do not depend on content scarcity.

IV. The Way Forward

A new technical paradigm for media reproduction and distribution is being established. As explained in [4], the scarce resource under exploitation should no longer be the information commodity but the user's attention. The earlier should be considered an investment that is made in order to acquire the later. The employed BMs must thus lean towards the free consumption of info goods, by the user community, laterally deriving gains from the context surrounding that activity, while taking full advantage of the techno-economical characteristics of on-line all-digital operation [4].

Consequently, a most adequate strategy is one of fostering a culture of proximity and interdependency, (already on the rise), between consumers, artists and CDists which enables the voluntary funding (e.g. Wikipedia, The Real News Network [13], or Radiohead's and Nine Inch Nails' experiments with voluntary user donations), of the two later entities by the earlier ones and facilitates other indirect revenue extraction means (e.g. advertisement, merchandising sale, live shows, etc) [4].

Commercial initiatives, that whish to thrive in on-line media delivery, must embrace the new paradigm. This will free their technical infrastructures from the excess of content access control tasks which they are presently responsible for. Said structures will then be able to operate in much freer and more innovative ways, enabling commercial P2P content delivery to succeed. These initiatives will then achieve the reliability that they have thus far been unable to attain – content delivery reliability, content governing reliability and economical reliability.

V. The P2PTube

A. Introduction

The P2PTube system is P2P based a content delivery

system, (under implementation), tailored to support BMs that are fully inline with the emerging paradigm in media reproduction and distribution.

Its architecture provides superior content delivery and security capabilities (to existing alternatives), through the employment of a set of innovative technical provisions such as: its hybrid P2P structure, which combines the security and coordinative capacities of a centralized system with the (reproduction and distribution) costs reduction properties of P2P content distribution; the integration of its robust security provisions with the hybrid content delivery structure which enables the exploitation of synergies between the two; and its user action monitoring capabilities

P2PTube thus enables an improved exploitation of the Internet medium's potential for costs reduction. It also enables the maintenance of a virtual social interaction medium, where the authenticity and/or integrity of all entities and contents is verifiable. This makes it possible for a secure exchange and consumption of media and user interaction and inter-rewarding to take place. All this facilitates a reliable lateral extraction of gains.

B. Business Model

The BM is the key aspect determining if an on-line media delivery initiative is inline with the emerging paradigm. P2PTube is meat to provide support to BMs, inline with such paradigm. For that reason it was conceived around a specific open access BM, simplistically, described bellow:

- consumer (and all other) users:
  - may consume any and all Media Items (MIs) made available by the system, free of charge;
  - load their accounts with currency which they may extract at any later time. That currency may be:
    - transferred directly from the user's bank account;
    - obtained by the user, by specifically watching commercial advertisements, which the system rewards;
  - perform donations to producer users they deem meritorious:
- producer users:
  - supply Media Items (MIs) to the system, for no immediate payment, specifying if they expect/accept to be rewarded (by the user community) for the delivery of such items;
  - obtain revenue from the user's voluntary donations to them;
- advertiser users:
  - supply advertising media items to the system, specifying the level of exposure they desire for them, in terms of its cost in the system's employed currency, which is calculated in accordance with a system defined price for user attention time;
  - funds are deducted from their account, by the system, as user's are exposed to their message;
- the system:
  - freely delivers all MI to all consumers;
  - rewards the watching of advertisements;
  - sells users' attention to advertiser users;
  - taxes all donation transactions;

Furthermore, given the system's P2P nature, it distributes the costs of content reproduction, storage and distribution throughout the user/peer community.

C. Architecture

P2P content delivery systems have typically evolved from a hybrid centralized/P2P operation mode (Napster), to more decentralized ones (Gnutella), especially in the non-commercial sector. This occurred for technical purposes (avoid an expensive central entity, and central point of failure), and to avoid the legal targeting of the central provisions of such systems (e.g. the shutting down of Napster). This, however, has resulted in a number of trust, content delivery efficiency and content discovery problems, among others. Therefore, as explained in section II.A.2, some operational centralization has been re-embraced, but maintaining a compromise between technical performance, global operation costs and legal targetability.

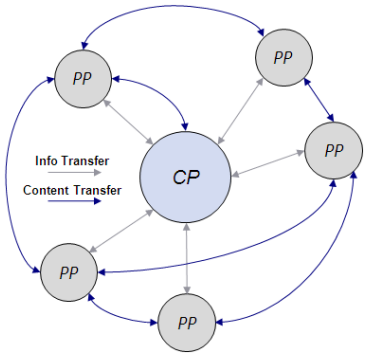

Figure 4. P2PTube Overall Structure

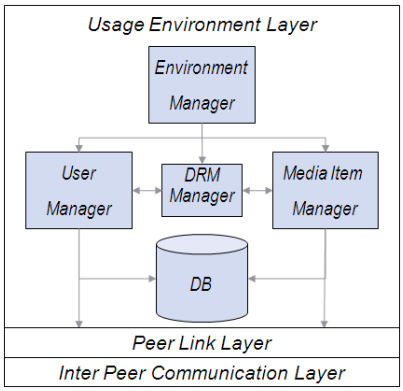

Figure 5. Peer Architecture

As P2PTube is designed to operate legitimately and to obtain revenue (that can be invested in the system's continuous upgarde), the optimal point of that compromise is necessarily different. It does not need to avoid having a central provision. As such, it employs a hybrid P2P structure [1], composed by a single coordinating Core Peer (cp), and any number of, Peripheral Peers (PP), as presented in Figure 4.

The cp , which is owned operated by P2PTube's CDist, runs on high capacity hardware, and coordinates the system. The PPs, which are owned and operated by regular users, run on household PCs and interface (host) the users with the system's interaction environment.

All system peers have the same internal structure (Figure 5). Each peer is divided into three layers, (and so is the system, as these layers traverse all the system's peers):

- The Inter-Peer Communication Layer (IPCL) handles the exchanging of information packets between different peers.
- The Peer Link Layer (PLL) secures all inter-peer communication and handles the resolution of peer contact endpoints.
- The Usage Environment Layer (UEL) handles user management and authentication operations, the original indexing, distribution, removal and discovery of MIs, the maintenance and validation or MI integrity and authenticity, the preservation and validation of MI authorship rights, the maintenance of user accounts, the secure performing of currency transactions, the maintenance of a coherent interaction medium and the interfacing of the users with that medium.

D. Operation

Each system layer develops its own independent internal operations. Depending on their requirements, some operations are conducted in a client/server manner while others are so in a P2P manner. The following sub-sections present the system's operation, in a *per-layer* basis, with a focus on security aspects.

1) IPCL Operation

The IPCL, operates in a purely P2P fashion. It provides the upper layers with communication services, handling the exchanging of information packets directly between peers.

2) PLL Operation

Every peer has a unique identifier, (p_i), and a unique asymmetrical key pair composed of a public key, (K_{p_i}), and a private key ($K^{-1}_{p_i}$), which it employs for its authentication. The ultimate register of the public part of this authentication information, (K_{p_i, p_i}), and of the contact endpoints of peers is the CP's PLL. It is assumed that the PLL of all peripheral peers knows cp's public key.

The PLL handles the registration, login and logoff of peripheral peers from the system (in a client/server manner), and the resolution of peer contact endpoints (in a hybrid manner). It also ensures the confidentiality, integrity, authenticity and non-repudiatiability of all messages exchanged between peers.

a) Peripheral Peer Registration

This process is performed in a client/server manner the first time a peripheral peer interacts with the system. The process, occurring at the PPL, is depicted in Figure 6 (taking into consideration the notation defined before that figure).

The registering peer, p_i, begins by (step 1), sending cp a message encrypted with the latter's public key (K_{cp}), so that only cp may read it (assuring confidentiality). The message's content's are signed by p_i, in order assure its integrity and authenticity of origin. The message, $regreqmsg^{cp}_{p_i}$, contains p_i's identification package signed by p_i, (sid_{p_i}),and a unique randomly generated nounce that cp will have to sign to prove its identity. sid_{p_i} is signed by p_i so that any system peer can assert that p_i validates that information (its thus can not be faked even by cp). It is signed independently from (and redundantly to) the overall message so that it may latter be independently redistributed.

cp then verifies the validity of the received message. If all is ok, it sends back (step 2) a challenge message to p_i in order to check if it truly possesses the private key corresponding to K_{p_i}, (that p_i specified in sid_{p_i}), as the message received in step 1 may be a repetition attack. The challenge message is encrypted with K_{p_i} so that only p_i may decipher it and its contents are signed by cp. It contains the first nounce (thus

u_i = a specific system user id
$psswd_{u_i}$ = password of u_i
$usname_{u_i}$ = the username of u_i
K_{u_i} = the public key of user u_i
$K^{-1}_{u_i}$ = the private key of user u_i

p_i = a specific peripheral peer id
K_{p_i} = the public key of p_i
$K^{-1}_{p_i}$ = the private key of p_i

cp = the core peer id
K_{cp} = the public key of cp
K^{-1}_{cp} = the private key of cp

o_i = a specific media item/objec t
$L^{o_i}_{u_i}$ = the license governing the use of o_i by u_i

$K^{s^{p_i-cp}_g}_s$ = a secret symmetric key for the encryption of inter - peer communicat ion during the g^{th} cooperatio n session between p_i and cp, $(s^{p_i-cp}_g)$

h(x) = cryptograp hic hash function applied to x
enc $_K$ (x) = the encryption of x with K
signed $_K$ (x) = enc $_K$ ($h(x)$)$\|x$ = signed x with key K
secmsg $_K$ (x) = enc $_K$ (x)

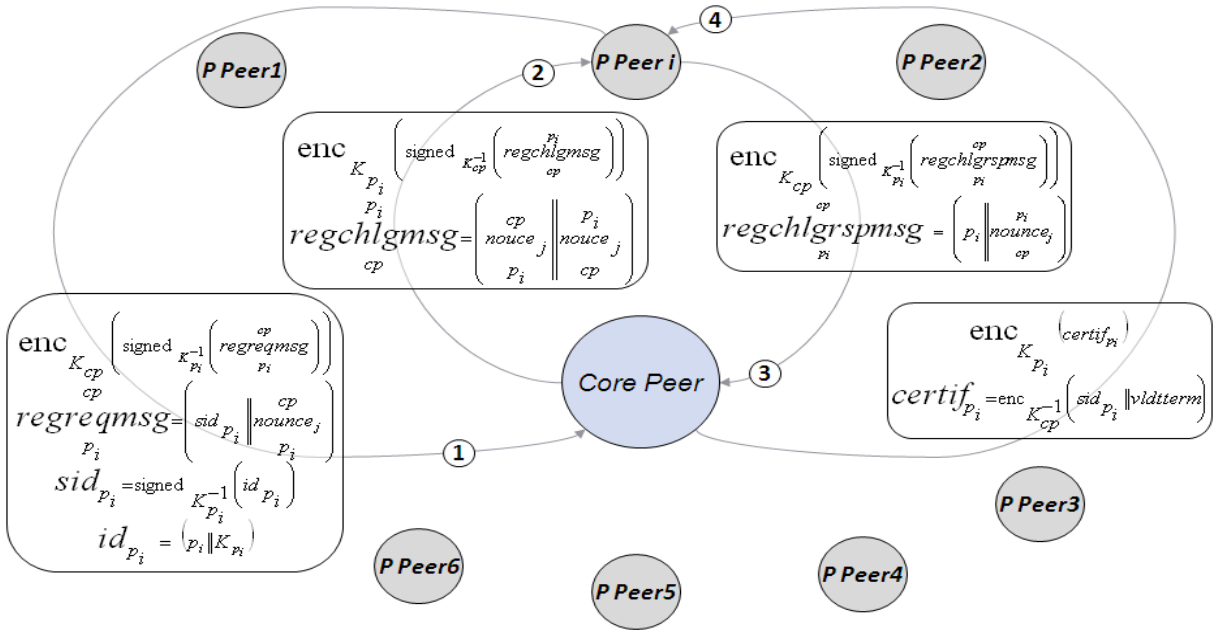

Figure 6. Peripheral Peer Registration Process

proving cp's identify), and a second new unique nounce that p_i will have to sign to fully prove its possession of $K_{p_i}^{-1}$.

p_i proceeds to sign the second nounce and send it back to cp (step 3). Finally cp validates the nounce's signature and, if all is ok, produces and signs the peer's registration certificate, $certif_{p_i}$, and sends it to p_i. This certificate contains sid_{p_i} and the validity term after which it expires. It may be freely exchanged amongst the peer community as a proof of p_i's and cp's mutual acceptance of p_i's participation in the system and of the latters public authentication credentials.

b) Communication Session Establishment

Before any two peers can communicate they must first establish a communication session between them. In the case of inter-peripheral peer communication, such a session is opened if two specific peripheral peers, p_i and p_k need to interact. The communication between peripheral peers and cp, is always necessary for a peer to participate in the system. Thus, whenever a registered peripheral peer, p_i, comes on-line, it must first establish a new (the g^{th} one) communication session, ($s_g^{p_i-cp}$), with cp, (the equivalent of loging into the system). This PLL process (p_i - cp communication case) enables the combination, between p_i and cp, of a secret symmetric key, ($K_s^{s_g^{p_i-cp}}$), to employ in the ciphering of all later communication between them within that session. Taking into consideration the notation defined in section V.D.2.a, this process, occurring at the PPL, is depicted in Figure 7.

This processes' initial step (step 1), is, in all, similar to the first step described in the registration process. The difference being that the sent message contains the peripheral peer's id (p_i), instead of its sid_{p_i}. cp validates the message and checks if its internal DB as any record of p_i. If so, it then proceeds to send, to p_i (step 2), a challenge message, in order to check the veracity of its claimed identity.

That message is encrypted with K_{p_i} so that only p_i may decipher it (assuring confidentiality), and its contents are signed by. It contains the nounce sent by p_i (thus proving cp's identify), and a second new unique nounce that p_i will have to sign to fully prove its identity. It also contains the new prospective session's id and the session key.

p_i proceeds to sign the second nounce and sends it back to cp(step 3). Finally cp validates the nounce's signature and, if all is ok the session is considered open. In the opposite case, event if the session id and key have already been sent, the session will not be valid, the key will be discarded and p_i will no be logged on.

The process of establishing an inter-peripheral communication session is similar to the one previously described. The only difference is that the involved peripheral peers will need the assistance of the cp to obtain each other's public keys, if they do not yet have them.

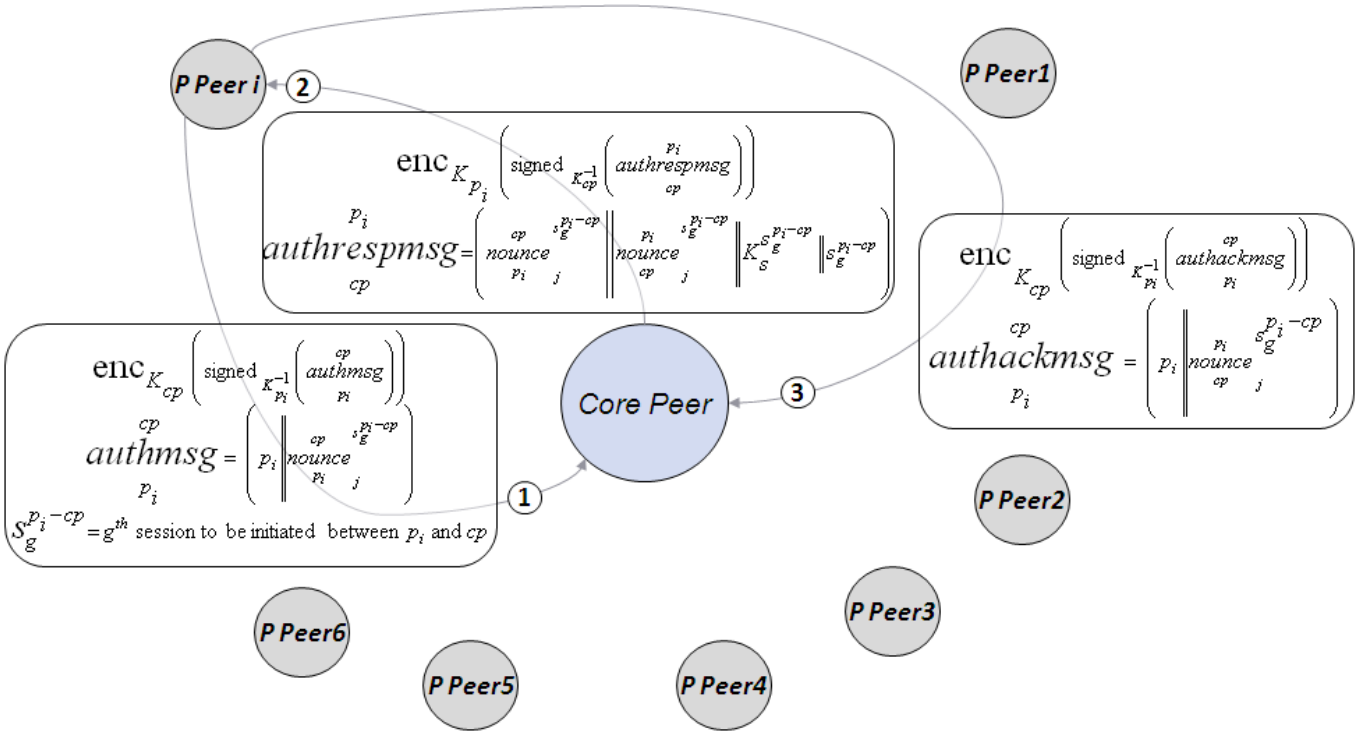

Figure 7. Peripheral Peer Login Process

c) Message Securing

The PLL assures the confidentiality, integrity, authenticity and non-repudiatiability of the messages exchanged between communicating peers.

In what regards the PP-CP communication, it does so in the manner depicted in Figure 8 (taking into consideration the notation defined in section V.D.2.a).

Assuming that a communication session has previously been established between p_i and cp, if the "client" peer is p_i, it sends a $\text{reqtrf}_{p_i}^{cp}(info_s, info)$ message, (step 1), to cp expressing its request. $info_s$ is the sensitive part of the information being transmitted and $info$ is the non-sensitive part. $info$ is signed with p_i's private key, to assure integrity, prove origin authenticity and ensure non-repudiatiability. $info_s$ is concatenated with a serial identifier, signed by p_i (for the above stated purposes), and ciphered with the secret session key ($K_s^{s_g^{p_i-cp}}$), so as to ensure its confidentiality. The purpose of the serial identifier is to uniquely differentiate all exchanged messages so that no two messages are ever alike. If a repetition is ever detected, it is either an error or a replay attack. For that reason the serial identifier contains a counter value ($msgcnt_{p_i\ j}^{cp\ s_g^{p_i-cp}}$ = the j^{th} message counter value sent from p_i to cp within session $s_g^{p_i-cp}$), indicating the number of messages sent from p_i to cp within the current session

Every peer keeps count of all the messages exchanged within every communication session, in both directions, so that they are able to detect any repeated incoming messages and to correctly produce the serial identifiers of the ones they send.

The serial identifier also contains the session id ($s_g^{p_i-cp}$), a timestamp indicating the overall message's creation date, and the id of the sending peer (p_i).

cp then sends the adequate reply in a response message (sent in step 2), which has much the same structure of the one received form p_i. The differences are that the signing key is cp´s private one, and the serial identifier is obviously a different one. It will be a two parts serial identifier. The first part is the serial identifier of the received request message. The second part is another serial identifier, (with the same structure as defined before), containing the count of messages sent from cp to p_i within the ongoing session. This way, the response message is differentiated and also associated to the request message to which its responds.

The communication procedure employed is the same as the one defined before if the communication is between two peripheral peers or, if the "client" peer is cp, with p_i playing the responder role (interactions 1a and 2a in Figure 8).

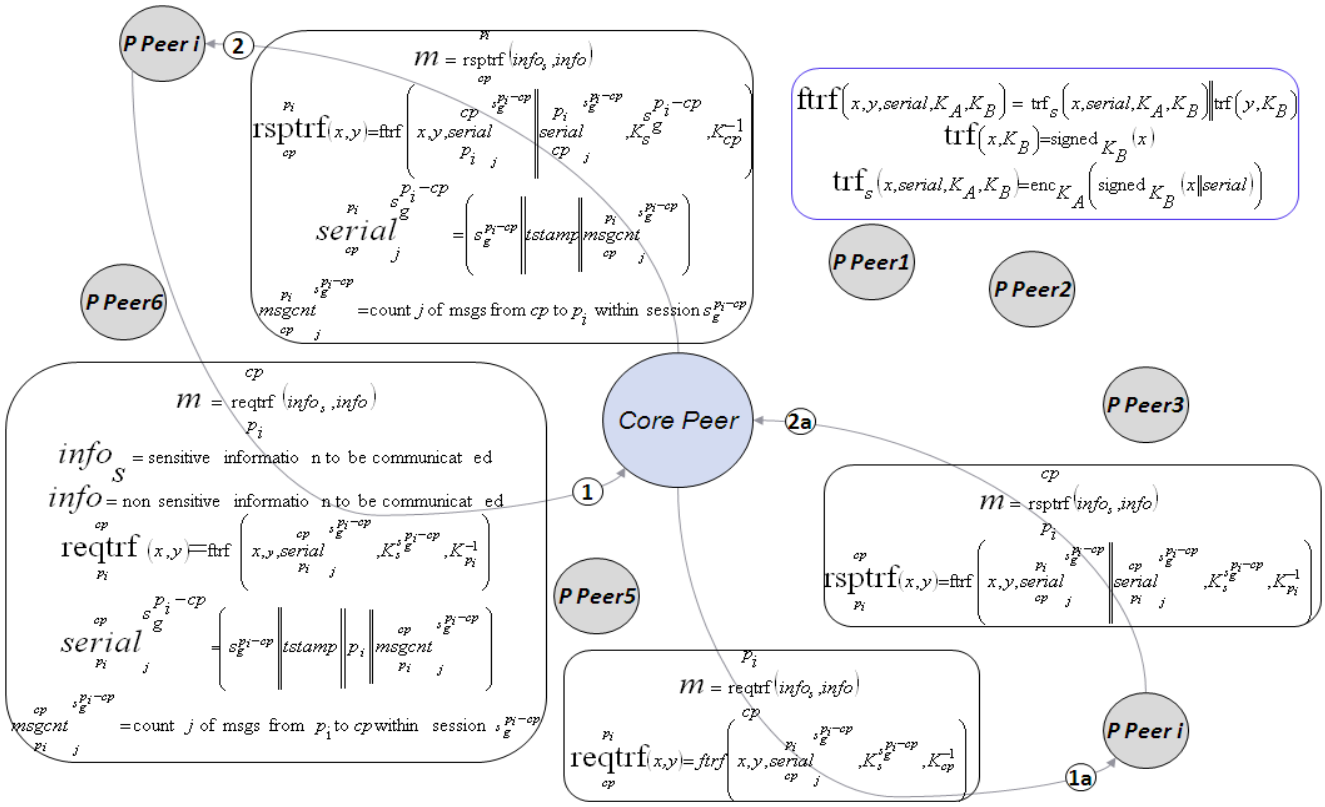

Figure 8. PP-CP Communication

d) Peer IP Discovery

This process is performed in a client/server or hybrid manner, whenever a peripheral peer p_i needs to know the IP of another peripheral peer p_k. Taking into consideration the notation defined in section V.D.2.a, this PLL process happens as follows.

p_i begins by sending, to cp, a request message ($\underset{p_i}{\overset{cp}{\text{reqtrf}}}\left(info_s, info\right)$), indicating that it wants the IP address of p_k. cp may immediately send back a package ($pploc_s = \text{signed}_{K_{cp}^{-1}}\left(p_k \| IP_k\right)$), signed by cp, containing that information or it may redirect p_i to some other peripheral peer, p_j, by sending p_i a signed retrieval permit ($retrpermit_{pploc_s}^{p_i} = \text{signed}_{K_{cp}^{-1}}\left(p_i \| p_j \| p_k\right)$), enabling it to retrieve the desired information from from p_j.

3) UEL Operation

The UEL handles, in a client-server manner all the operations related to:

- the registration, login and logoff of users – employing the User Manager (*UMngr*) components (see Figure 5);
- the collection of information on user actions, the maintenance of user accounts and performing of currency or credits transactions – employing the *UMngr* and/or DRM Manager (*DRMMngr)* components;
- the injection or removal of MIs – employing the Media Item Manager (*MIMngr*) and/or *DRMMngr* components;

In all such cases, cp's UEL plays the server role and the involved PP's UEL, plays the client role. The client peers will typically send their requests to the server side, which judges their legitimacy and, if adequate, enforces them or provides some required information, if that is the case.

A mixed P2P and client/server operation mode is employed, by UEL, in:

- the semantics based searching for MIs;
- the retrieval/distribution of MIs;
- the searching and retrieval/distribution of information objects pertaining to users, peers or other aspects of the system.

The core peer's UEL instance holds all the information pertaining to the location of individual MIs in the system's (peripheral) tissue, as well as the MI semantic characterization information, against which user queries must be matched for content searches. It also holds all other relevant information.

Whenever such information is needed, peripheral peers will request it from the cp, which may respond by handing it over directly or by informing the "client" PP about appropriate "server" PPs for the acquisition of the desired information.

a) User Registration

This UEL process is performed in a client/server manner and is depicted (taking into consideration the notation defined in section V.D.2.a), in Figure 9.

The registering user, (identified as u_i), delivers, to her hosting peer, (p_i), her public identification package, id_{u_i}, (containing u_i, $uname_{u_i}$, and the user's public key, K_{u_i}), and also her private authentication components $passw_{u_i}$, and private key, $K_{u_i}^{-1}$ (step 1). p_i signs id_{u_i} with $K_{u_i}^{-1}$ (to prove, without repudiability, that u_i approves that information), and sends it, together with $passwd_{u_i}$, to cp (step 2). That information is sent as the sensitive part of a $\underset{p_i}{\overset{cp}{\text{reqtrf}}}$ message (see section V.D.2.c), thus assuring its confidentiality, integrity and origin authenticity. If all is ok cp registers the new user and produces a signed certificate validating the user's participation in the system ($certif_{u_i}$), which it sends (step 3), to p_i as the sensitive part of a $\underset{cp}{\overset{p_i}{\text{rsptrf}}}$ message. Finally p_i delivers $certif_{u_i}$ to u_i (step 4).

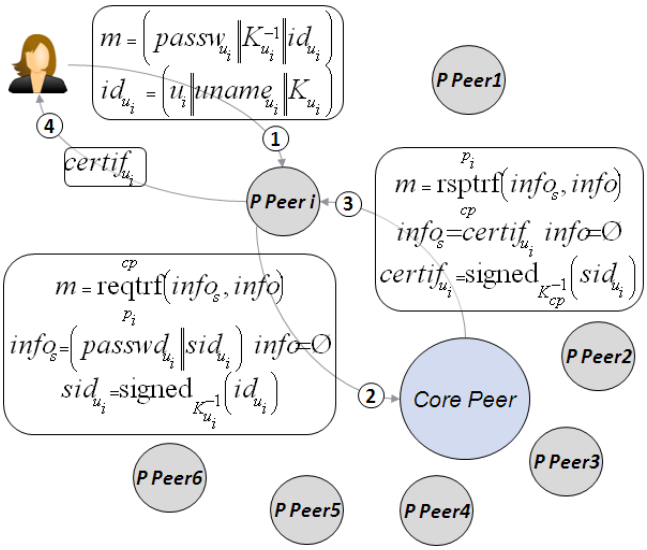

Figure 9. User Registration Procedure

b) User Light Authentication

The purpose of a light authentication is to enable, an already registered user, to login, without having to supply his key pair to his hosting peer, and gaining access to only some of the system's services. The user will only be allowed to perform actions which do not imply any change or addition to the system's data (typically only MI searches and consumptions).

This UEL process is performed in a client/server manner and is depicted (taking into consideration the notation defined in section V.D.2.a), in Figure 10.

The user supplies her light authentication data (username and password) to her hosting peer, p_i (step 1). p_i executes a specific hash function, $f(lauthdata_{u_i})$, on that information (employment of digest access authentication), and sends it in a secure fashion to cp (step 2). If all is ok, cp prepares a signed (by itself), user hosting certificate $hostcertif_{u_i}^{p_i}$. It contains sid_{u_i}, p_i and a timestamp. It demonstrates the system's acknowledgment that p_i is hosting u_i up to the time instant specified by the timestamp. After that time, p_i must again login u_i. $hostcertif_{u_i}^{p_i}$ is then securely sent to p_i (step 3).

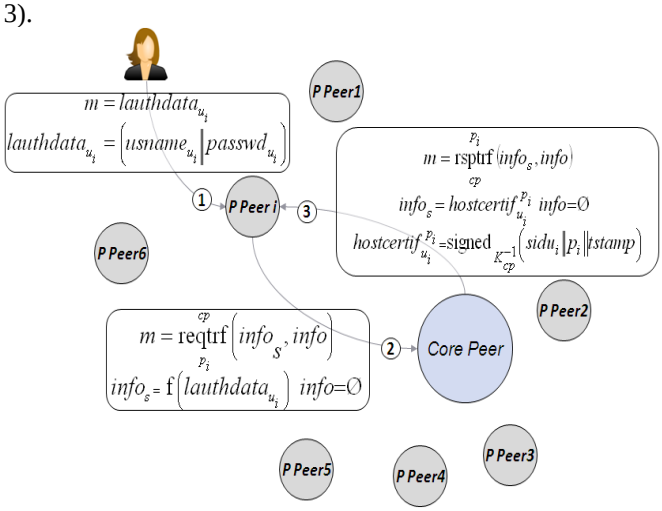

Figure 10. User Light Authentication Procedure

c) User Full Authentication

A full authentication enables the user to make use of all the system's services. This UEL process is performed in a client/server manner and is depicted (taking into consideration the notation defined in section V.D.2.a), in Figure 11.

u_i delivers her full authentication data to her hosting peer, p_i (step 1). That package contains the user's password, username and key pair. p_i communicates with cp (step 2), indicating that it wishes to do a full login of u_i. cp responds (step 3), with a nounce that p_i will have to sign with u_i's private key to prove that it is in fact hosting u_i. p_i (in the employment of digest access authentication), executes a specific hash function on $(uname_{u_i} \| passw_{u_i})$, and then returns to cp, (step 4), a signed package containing the calculated hash value and the received nounce, thus proving that it is indeed hosting u_i. If all is ok cpthen returns (step 5) a similar user hosting certificate to the one described in the previous section.

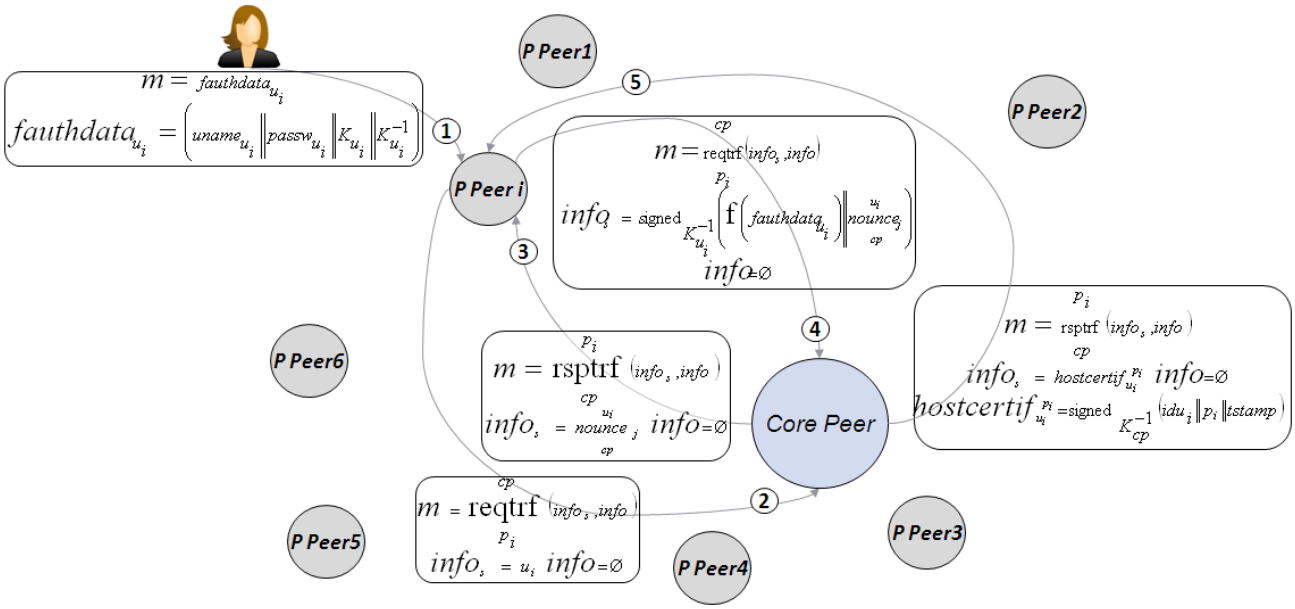

Figure 11. User Full Authentication Procedure

d) Further Client/Server User Attending Operations

As mentioned the maintenance of user accounts and performing of currency or credits transactions as well as the injection of MIs are handled in a client/server manner at the UEL. The typical system procedure to handle such user request is depicted in Figure 12, (taking into consideration the notation defined in section V.D.2.a).

u_i delivers the sensitive and non-sensitive input parameters that make up its request, to his hosting peer, p_i (step 1). p_i then signs both parameters with u_i's private key (for origin authenticity proof and non-repudiability assuring), concatenates each of them with u_i (so that the receiving end

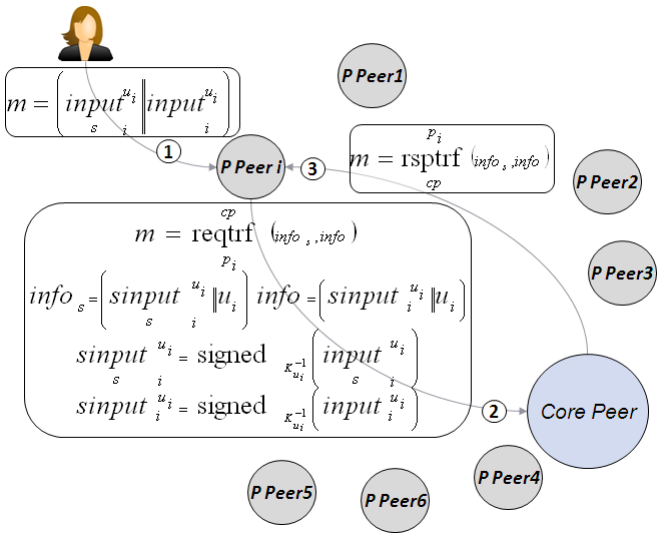

Figure 12. Client/Server User Attending Procedure

may know which public key to employ to validate the signatures), and securely sends them to cp (step 2).

cp produces the adequate response and securely sends it to p_i (step 3), which displays the relevant output to u_i.

For instance, if this was an MI injection operation, the non-sensitive input parameter, $input_i^{u_i}$, would be the MI object, identified as o_i. This means that it would be sent to cp, signed by u_i, but not encrypted with the session key, as this would be resource consuming and not security-wise relevant. o_i's signature would prove that u_i, and not anyone else, did in fact submit it. If all was ok, the cp would accept o_i, and notify p_i of this. It would then sign o_i, with its own private key, to signal the system's acceptance of that objects injection into it, thus producing o_i's delivery-ready version, o_i^{cp}. Finally cp would perform the initial seeding of o_i^{cp}, through the system's tissue, by sending it for storage and redistribution to some selected peripheral peer.

e) Hybrid Operations

As stated in section V.D.3, a mixed P2P and client/server operation mode is employed, by UEL, in operations such as the semantics based searching for MIs and the retrieval/distribution of MIs among others. The way these operations unfold is similar. To explain them we present bellow the MI retrieval operation.

The typical system procedure to retrieve a specific MI object, o_i^{cp}, is depicted in Figure 13, (taking into consideration the notation defined in section V.D.2.a).

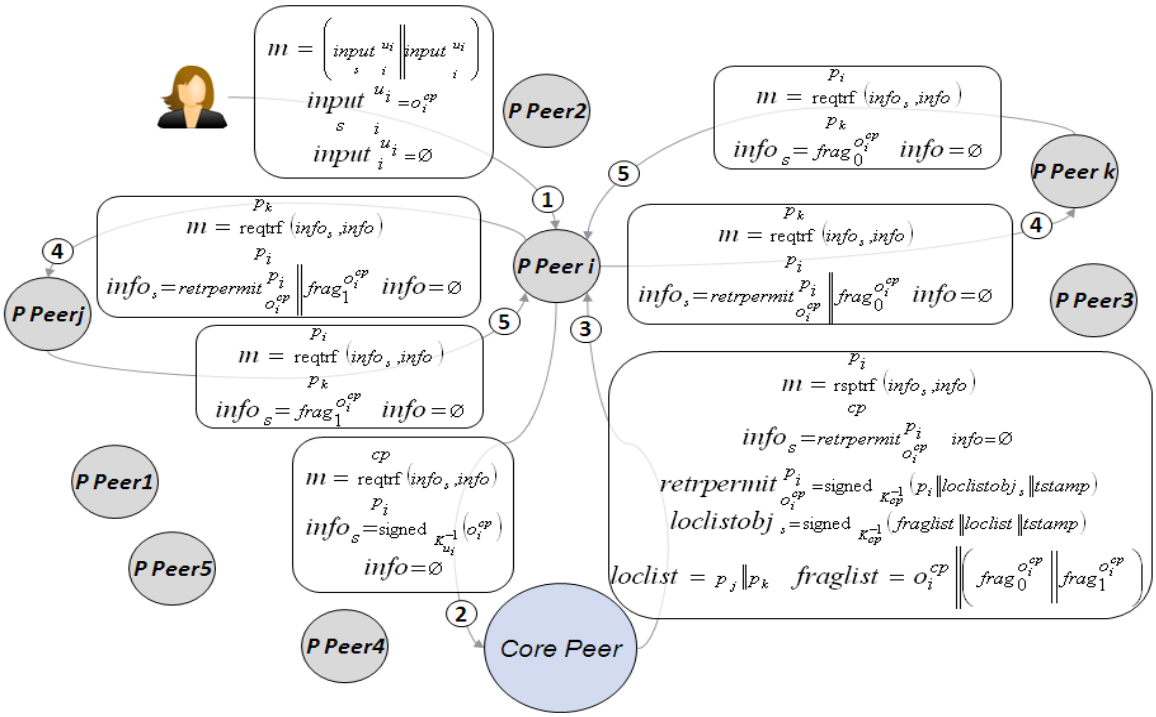

Figure 13. Hybrid Operation for MI retrieval

u_i informs it hosting peer, p_i, that she wishes to consume MI, o_i^{cp} (step 1). p_i signs the request information with u_i's private key (establishing its origin), and securely sends the signed request to cp (step 2). cp prepares an MI retrieval permit ($retrpermit^{p_i}_{o_i^{cp}}$), enabling p_i to perform the retrieval of o_i^{cp} from a number of other peripheral peers. The permit contains the identification of the peer to which it has been emitted (p_i), the $loclistobj_s$ object, and a timestamp indicating the permit's validity term. $loclistobj_s$, is signed by cp, and carries a list of the peers from which o_i^{cp}, may be retrieved, the *fraglist* object and a timestamp indicating the time at which $loclistobj_s$ was compiled. *fraglist* contains the MI's id (o_i^{cp}) and the identification of the fragments into which o_i^{cp} has been subdivided to enable it fragmented redistribution. $retrpermit^{p_i}_{o_i^{cp}}$ is signed by cp to assure its authenticity.

The independent signing of $retrpermit^{p_i}_{o_i^{cp}}$ and $loclistobj_s$ enables the later independent P2P distribution of $loclistobj_s$ amongst the peripheral peers.

cp then sends $retrpermit^{p_i}_{o_i^{cp}}$ to p_i (step 3). After that, p_i proceeds (in a P2P fashion), to simultaneously retrieve o_i^{cp}'s composing fragments, from its delivering peers (e.g. p_j and p_k). It sends messages to p_k and p_j (steps 4), requesting fragments $frag_0^{o_i^{cp}}$ and $frag_1^{o_i^{cp}}$ respectively. In such messages it includes the permit received from cp, so that the p_k and p_j can verify that p_i is indeed authorized to retrieve, from them, the requested MI. Finally p_k and p_j deliver said fragments (steps 5), to p_i which reconstitutes o_i^{cp}, validates its integrity and renders it for user consumption.

This procedure thus enables an efficient P2P diffusion, of MIs, between peripheral peers. It may also be applied to other information objects as user participations certificates ($certif_{p_i}$), user hosting certificates, ($hostcertif^{p_i}_{u_i}$), content search response objects (see *SQRO* in section V.E.a), or MI location describing objects ($loclistobj_s$).

f) User Action Monitoring

User treatment of different MIs will vary. After having retrieved an MI, from the system, for local consumption, the consuming user, u_i, may for instance, watch it numerous times, only once or not at all. These different consumption behaviors indicate different preferences on the part of users.

Information pertaining to such user behaviors may, for instance, be employed for directed marketing or targeted advertising. Thus, in order to provide further support to the lateral extraction of gains, P2PTube also provides functionalities for the gathering of information on user behavior.

Operations regarding the gathering and submission of this type of data, are performed at the peripheral peers (as these are the ones hosting the users), but are requested, (to the later), by the cp. The typical procedure is the following.

Whenever cp needs such information about user u_i, hosted at p_i, it sends, (to p_i), a signed Event Report Request (ERR), soliciting the monitoring of specific events on the part of u_i. p_i validates the ERR. If all is ok, p_i sends a signed acknowledgment to cp and proceeds to perform the requested monitoring.

If and whenever the targeted user action occurs, p_i prepares the corresponding Event Report (ER), signs it and sends it to cp.

E. Data and Metadata Objects

The regular operation of the system involves the production and exchange/distribution of a set of different information objects. To assure systemic reliability, these must be structured so as to guarantee their integrity, authenticity and temporal validity.

In P2PTube, MPEG-21 is employed for the structuring of such objects, given the security-wise and information structuring capabilities of that standard. The most relevant of these objects are presented bellow.

a) Search Query Response Objects

```
<DIDL xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS"
      xmlns:sqro="urn:p2pt:sqro"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
      xsi:schemaLocation="urn:mpeg:mpeg21:2002:02-DIDL-NS
      didl.xsd">
  <Container id="sqro">
    <Descriptor id="sqro_signature">
      <Statement mimeType="text/xml">
        <dsig:Signature>
          ....
        </dsig:Signature>
      </Statement>
    </Descriptor>
    <Item id="sqro_milist">
      <Descriptor id="sqro_milist_emissiontime">
        <Statement mimeType="text/plain">
          "YYYY-MM-DDThh:mmTZD"
        </Statement>
      </Descriptor>
      <Descriptor id="sqro_milist_answeredquery">
        <Statement mimeType="text/xml">
          ....
        </Statement>
      </Descriptor>
      <Component>
        <Resource mimeType="text/xml">
          <sqro:MIList>
            ....
          </sqro:MIList>
        </Resource>
      </Component>
    </Item>
  </Container>
</DIDL>
```

Figure 14. SQRO Example

Before a user decides to consume a specific media item (causing the operation exemplified in section V.D.3.e), he must first learn about the existence of such an object. That knowledge is typically obtained through content searches (e.g. googling).

The user will thus specify a number of semantic criteria that his target MI(s) must respect and tell his hosting peer to obtain a list with such MIs. The searching procedure that will then ensue, will unfold in the same hybrid manner as the operation presented in section V.D.3.e. This way the host peer will ultimately receive an object (file), containing the response from cp or another peripheral peer (under cp indication).

This Search Query Response Object (*SQRO*), is expressed as an MPEG-21 DID [33] (example depicted in Figure 14), with the following structure:

- an inner *did:Item* – it carries:
  - the emission date of the query response information – within *did:Descriptor* "*sqro_milist_emissiontime*";
  - the answered query – within *did:Descriptor* "*sqro_milist_answeredquery*";
  - the actual query response information – within the "*sqro:MIList*" child of its *did:Component* child;
- an outer *did:Container* – it carries:
  - the inner *did:Item*;
  - the security assuring provisions – consist of the digital signature, (by cp), of the inner *did:Item*, expressed by a *dsig:Signature* element of *did:Descriptor* "*sqro_signature* ".

The *sqro:MIList* element is structured in accordance with the schema depicted in Figure 15.

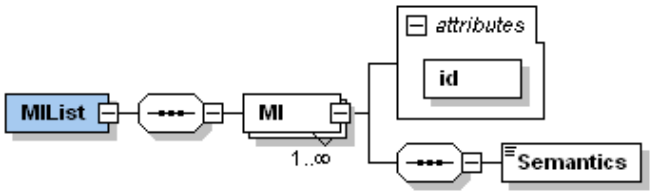

Figure 15. MIList Schema Depiction

The signing of the inner item, (by cp, at the time of cp's original production of the *SQRO*), assures *SQRO*'s integrity and origin authenticity during its propagation across the system's periphery, thus enabling its P2P diffusion, independently of the cp.

SQRO's emission date, contained within the "*sqro_milist_emissiontime*" *did:Descriptor*, enables any peripheral peer to assess the freshness or staleness of the information.

b) MI Location Describing Objects

Object $loclistobj_s$, is simplistically described, in section V.D.3.e, as signed $_{K_{cp}^{-1}}\left(fraglist \| loclist \| tstamp \right)$. In more precise terms it consists of an MPEG-21 DID with the following structure (example depicted in Figure 16):

- an inner *did:Item* – it contains:
  - a list of peripheral peers available to deliver the MI – within the child *milo:PPList* element of *did:Descriptor* "*milo_loclist*";
  - the identification of the MI at stake – as the value of the ref attribute of the did:Resource child element of the *did:Component*;

- the specification of the fragments into which the MI is divided. For each fragment there will be a *did:Anchor* element (inside the *did:Component*), which carries the identification of the fragment as its id, and carries the fragment's definition within a *did:Fragment* element [31].

- a middle did:Item containing:
  - the emission time of $loclistobj_s$ – within *did:Descriptor* "*milo_emissiontime*";
  - the inner *did:Item*;
- an outer *did:Container* – it carries:
  - the middle *did:Item*;
  - the digital signature, (by cp), of the middle *did:Item*, expressed – within *did:Descriptor* "*milo_cp_signature*".

```
<DIDL xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xsi:schemaLocation="urn:mpeg:mpeg21:2002:02-DIDL-NS didl.xsd">
  <Container id="milo">
    <Descriptor id="milo_cp_signature">
      <Statement mimeType="text/xml">
        <dsig:Signature>.....</dsig:Signature>
      </Statement>
    </Descriptor>
    <Item id="milo_middle">
      <Descriptor id="milo_emissiontime">
        <Statement mimeType="text/plain">
          "YYYY-MM-DDThh:mmTZD"
        </Statement>
      </Descriptor>
      <Item id="milo_inner">
        <Descriptor id="milo_loclist">
          <Statement mimeType="text/plain">
            <milo:PPList>
              .....
            </milo:PPList>
          </Statement>
        </Descriptor>
        <Component>
          <Resource mimeType="application/mi" ref="targetMI_ID"/>
          <Anchor id="frag_0">
            <Fragment fragmentId="offset(0,100000)"/>
          </Anchor>
          <Anchor id="frag_1">
            <Fragment fragmentId="offset(100000,200000)"/>
          </Anchor>
        </Component>
      </Item>
    </Item>
  </Container>
</DIDL>
```

Figure 16. $loclistobj_s$ example

The signing of the middle item by cp enables the validation of the $loclistobj_s$'s integrity and of its origin authenticity.

c) Media Items Objects

P2PTube's MIs consists of TAR archives which have the following content:

- MI Head File (*MIHFile*) – this file contains all of the MI's metadata;
- MI Content File(s) (*MICFile*) – each containing an actual media (video) content;

```
<DIDL xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
      xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
      xsi:schemaLocation="urn:mpeg:mpeg21:2002:02-DIDL-NS
      didl.xsd">
  <Container id="mih">
    <Descriptor id="mih_cp_signature">
      <Statement mimeType="text/xml">
        <dsig:Signature>.....</dsig:Signature>
      </Statement>
    </Descriptor>
    <Item id="mih_mid">
      <Descriptor id="mih_mid_u_signature">
        <Statement mimeType="text/xml">
          <dsig:Signature>.....</dsig:Signature>
        </Statement>
      </Descriptor>
      <Item id="mih_mid_inner">
        <Descriptor id="mih_mid_inner_semantics">
          <Statement mimeType="text/xml">
            <rdf:RDF>.....</rdf:RDF>
          </Statement>
        </Descriptor>
        <Descriptor id="mih_mid_inner_rights">
          <Statement mimeType="text/xml">
            <r:license>.....</r:license>
          </Statement>
        </Descriptor>
        <Item id="base_item1">
          <Descriptor id="base_item_signature">
            <Statement mimeType="text/xml">
              <dsig:Signature>.....</dsig:Signature>
            </Statement>
          </Descriptor>
          <Component>
            <Resource mimeType="video/mpeg" ref="video1.mpg"/>
          </Component>
        </Item>
      </Item>
    </Item>
  </Container>
</DIDL>
```

Figure 17. *MIHFile* Example

This *MIHFile*, carries an MPEG-21 DID (example depicted in Figure 17), with the following structure:

- an inner *did:Item* – it contains:
  - the semantically qualifying information for the MI – within *did:Descriptor* "*mih_mid_inner_semantics*";
  - the rights information pertaining to the MI – REL element *rel:license* [8], within *did:Descriptor* "*mih_mid_inner_rights*";
  - a *did:Item* element for the declaration of each of the *MICFiles* in the MI. Each such *did:Item* contains:
    - the digital signature, (by the owner user u_i), of the corresponding *MICFile* – within *did:Descriptor* "*base_item_signature*";
    - a *did:Resource* element referencing the target *MICFile*;
- a middle *did:Item* containing:
  - the digital signature, (by u_i), of the inner *did:Item* – within *did:Descriptor* "*mih_mid_u_signature*";
  - the inner *did:Item*;

- an outer *did:Container* – it carries:
  - the middle *did:Item*;
  - the digital signature, (by cp), of the middle *did:Item*, expressed – within *did:Descriptor* "*mih_cp_signature*".

The signing, by u_i, of each of the MI's *MICFiles* enables the validation of their integrity and of their origin authenticity. It also guarantees their non repudiability. Furthermore, the signing, by u_i, of the inner item enables the same actions towards the inner Item, thus securing the semantic and rights metadata.

The signing, by cp , of the middle item proves that the system has effectively accepted the insertion of the MI.

In accordance with what was expressed in section V.D.3.d, at the production time of an MI, o_i, u_i produces the signatures of all *MICFiles*, prepares the inner item declaration, produces the signature of the latter item and then builds the middle and outer item declarations, placing them in the *MIHFile* which it concatenates with the *MICFiles*. This way u_i produces a o_i which is effectively signed by him.

o_i is sent to cp which unpackages it, calculates the signature of the middle item and inserts it in the outer item. It then repackages o_i into what then is its distribution ready form o_i^{cp}.

d) User Monitoring Requests and Responses

The ERR and ER objects mentioned in section V.D.3.f are MPEG-21 DIDs carrying MPEG-21 ERRs and MPEG-21 ERs respectively [32].

VI. Comparison

When compared, with the systems and initiatives presented in section II, P2PTube presents considerable advantages in terms of its overall reliability – delivery reliability, security reliability and economical reliability.

In what regards the non-commercial systems, P2PTube is much more reliable in the discovery and retrieval of content. Its central registry, (at the cp), of all the systems MIs, enables content searches to be performed over the entire set of available MI in a rapid manner without flooding the network with queries and in a single inter-peer interaction (between a peripheral peer and cp). Content retrieval is also optimized because cp has a global view of the availability for MI redistribution by the peripheral peer collective. cp directs downloading peers to the most appropriate other peers, thus performing an optimized load balancing and eliminating free riding.

The system's central structure also enables it to provide a number of security facilities which are generally absent from the non-commercial sector, such as guaranteeing the confidentiality, authenticity, integrity and non-repudiability of all exchanged messages, guaranteeing the authenticity, integrity and non-repudiability of all the MI, and other information objects exchanged across the system (either in a client/server or P2P fashion), and guaranteeing peer, user and MI identity. P2PTube is thus capable of supporting a secure management and transaction of monetary funds.

Even if compared to typical non-commercial P2P systems employing distributed collective/mutual authentication solutions, as those mentioned in section II.A.2, P2PTube is at an advantage because such solutions:

- imply a considerable operational overhead – multiple interactions between peers are necessary to achieve some security. In P2PTube a more robust level of security is achieved, with far less interactions;
- present considerable weaknesses in the face of sufficiently vast and attacks – require that a minimum quorum of "honest" peers is present. No such necessity in P2PTube;
- frequently assume the reliability of the network – no such assumption in P2PTube;

It may be argued that, as P2PTube is coordinated by a central entity, it is less scalable than the typical non-commercial solution. This is not so. Fully distributed P2P systems present a vast range of scalability problems itselves, related to the inter-discovery of peers and to the discovery of content, as the number of peers in the system grows and no entity exists with a global and consistent view of the system's state. In P2PTube, cp eliminates such problems, and, if enough resources are invested into cp it's centrality will not be a problem, just like no problem is posed by the centrality of Wikipedia's and Google's central server farms.

For all this, this system is capable of reliably supporting a number of business models, for the delivery of media content, which the non-commercial systems clearly cannot.

In what regards the commercial systems, approached in section II, P2PTube presents several technical and economic advantages.

P2PTube is tailored to support BMs that derive gains in a lateral manner to the actual consumption of media goods, (through advertisement and donations). Therefore, it does not have to "fabricate" content scarcity or enforce access control.

This enables the system to maintain a simpler and more efficient operation, as it does not require the intricate DRM provisions implied by content access restriction. Instead the system needs only to assure the confidentiality integrity and authenticity of communications and the integrity and authenticity of the MIs (and other data objects) which are diffused through it. This way, P2PTube provides a level of security which is, contextually, better than those of systems presented in section II.B, as it has simpler and more realistic security ambitions (than, for instance, Qtrax or ReelTime), which it achieves with greater solidity and efficiency.

Also, the P2PTube system does not rely on any pre-existing P2P structure (such as Qtrax and iMesh do), which globally is out of the system's control and presents reliability issues. Instead it employs a hybrid structure of its own. Content searching and retrieval are performed in a P2P fashion, but under the optimizing coordination of the system's, reliable,

central provisions, which also handle all trust and security demanding tasks and maintain a global and updated view of the system's overall state. This hybridness allows P2PTube to maintain a more efficient performance in terms of content searching and retrieval, than most systems presented in section II.B (e.g. Babelgum and Joost), but simultaneously, it does so in a more reliable way than most such systems (e.g. Qtrax and iMesh).

Furthermore, the full integration between the system's security and content delivery provisions enable it to take advantage of synergies between the two. It thus transforms the typical TTP security solutions, employed by commercial systems, into an integral component of the overall system which is able to exploit its P2P capabilities also for security purposes.

For all of the above, P2PTube is much better equipped, than the systems in scope, to take full advantage of data super-distribution, in a reliable manner.

P2PTube, in comparison to existing commercial systems, presents superior content delivery capabilities, powerful content usage monitoring provisions and more flexible security measures , which it focuses on assuring confidentiality, integrity and authenticity of messages, entities and data, instead of on content access restriction (as ReelTime and iMesh do). It thus provides more adequate tools, than the existing commercial systems, to reliably support the opened BMs that we deem necessary (section IV) for a successful operation in the field of P2P on-line content delivery.

VII. Conclusion

The field of information production, distribution, and consumption is rapidly changing. The technical, economic and social factors which shape it, are inexorably pushing it towards an ever more collaborative production mode, and an increasingly freer exchange of information goods.

To overcome the technical challenges that on-line content delivery presents, CDists must first overcome those of economic nature. Thus, they should simply embrace the ongoing evolution, by radically changing their business and operational paradigm. They should strive to maximize their content's exposure and social impact, and to supply the consumer-creator community with captivating virtual social bonding spaces where free content consumption and lateral revenue extraction may take place.

The P2PTube architecture provides superior tools (to existing alternatives), to reliably support BMs which are inline with the ongoing evolutive trend. Its contribution lies precisely in its identification of the reliable BMs for on-line content delivery, and its tailoring, for such BMs, through an innovative joint employment of a set of technical provisions, such as its hybrid P2P structure, the integration of its robust security provisions with the hybrid content delivery structure and its monitoring capabilities.

Given its characteristics, P2PTube allows its operating CDist to fully harness Internet era's economic potentials.

Future work on P2PTube's architecture should focus on transforming the cp into a distributed collective of core peers, which cooperate amongst themselves, employing a "web of trust" approach for the maintenance of trust and security. This evolution will render the system more efficient.

Acknowledgment

The authors would like to thank the Fundação para a Ciência e a Tecnologia for its financial support to the PhD studies within which this work was developed, and also INESC TEC for hosting those PhD activities.

References

[1] S. Androutsellis-Theotokis, D. Spinellis, "A survey of peer-to-peer content distribution technologies", *ACM Computing Surveys*, vol. 36, issue 4, pp. 335-371, December 2004.

[2] Babelgum Homepage, 2011, http://www.babelgum.com.

[3] E. Bangeman, "P2P traffic shifts away from music, towards movies", *Ars Technica*, http://arstechnica.com/tech-policy/news/2007/07/p2p-traffic-shifts-away-from-music-towards-movies.ars.

[4] H. Castro, A. P. Alves, C. Serrão, B. Caraway, " A New Paradigm for Content Producers", *IEEE MultiMedia*, vol. 17, nº 2, pp. 90-93, Apr. 2010.

[5] iMesh Homepage, 2011, http://www.imesh.com.

[6] Joost Homepage, 2011, http://www.joost.com.

[7] J. Maguire, "Hitting P2P Users Where It Hurts", *WiredNews*, 2003 http://www.wired.com/entertainment/music/news/2003/01/57112.

[8] MPEG-21 Consortium, "MPEG-21 Information Technology — Multimedia Framework — Part 5: Rights Expression Language, ISO/IEC JTC 1/SC 29/WG 11", ISO, 2003.

[9] OECD, "Organization for Economic Co-operation and Development Information Technology Outlook", 2006.

[10] PPLive Homepage, 2011, http://www.pplive.com/en/index.html.

[11] Qtrax Homepage, 2011, http://qtrax.com.

[12] Veoh Homepage, 2011, http://www.veoh.com.

[13] The Real News Network Homepage, http://www.therealnews.com.

[14] J. Brodkin, "Appeals Court reaffirms DMCA protection for user-generated content", *Ars Technica*, 2011, http://arstechnica.com/tech-policy/news/2011/12/appeals-court-reaffirms-dmca-protection-for-user-generated-content.ars.

[15] R. Andrews, "Joost Rival Babelgum Opens Doors To P2P TV Viewers", *paidContent*, 2007, http://www.paidcontent.org/entry/419-joost-rival-babelgum-opens-doors-to-p2p-tv-viewers.

[16] Screendigest, "Babel Networks to join increasingly crowded online video market", *Screendigest*, http://www.screendigest.com/online_services/intelligence/broadband/updates/bi-070307-dcsj1/view.html.

[17] G. Sandoval, "Skype founders name new video start-up Joost", *News.com*, 2007, http://www.news.com/Skype-founders-name-new-video-start-up-Joost/2100-1026_3-6150225.html.

[18] S. D. Kramer, "Joost Says It Has No Future As Portal, Enters White-Label Market; Volpi Out As CEO", *paidContent*, 2009,

http://paidcontent.org/article/419-joost-admits-no-future-as-portal-volpi-out-as-ceo-staff-cuts-white-labe.

[19] E. V. Buskirk, "Surprise! Qtrax, The 'Free and Legal Music Downloads' Service, Is Back", *Wired Magazine*, 2011, http://www.wired.com/epicenter/2011/03/qtrax-is-back

[20] A. Allen, "iMesh Inc. Subsidiary MusicLab Announces Beta Launch of BearShare Authorized Peer-to-Peer Service; Offering Includes ToGo Portable Music Subscription Service and Social Networking Features", *BusinessWire*, 2006, http://www.businesswire.com/news/home/20060817005185/en/iMesh-Subsidiary-MusicLab-Announces-Beta-Launch-BearShare.

[21] N. McKay, "Peer-to-Peer Goes Legit", *Wired Magazine*, 2005, http://www.wired.com/entertainment/music/news/2005/11/69457.

[22] Gnutella Protocol Development Registration Site, Sourcefourge, http://rfc-gnutella.sourceforge.net.

[23] The BitTorrent Protocol Specification, http://www.bittorrent.org/beps/bep_0003.html.

[24] A. Jantunen, S. Peltotalo, J. Peltotalo, "Peer-to-Peer Analysis State-of-the-art", Tampere University of Technology, 2006,

[25] Kazaa Media Desktop, http://www.kazaa.com.

[26] S. Garfinkel, *PGP: Pretty Good Privacy*, O'Reilly & Associates, Inc., Cambridge, MA, 1995.

[27] N. Saxena, G. Tsudik, J. H. Yi, "Threshold cryptography in P2P and MANETs: The case of access control", *Computer Networks*, vol. 51, nr. 12, pp. 3632-3649, August 2007.

[28] V. Pathak, L. Iftode, "Byzantine fault tolerant public key authentication in peer-to-peer systems", *Computer Networks, Special Issue on Management in Peer-to-Peer Systems: Trust, Reputation and Security*, vol. 50, nr. 4, pp. 579–596, March 2006.

[29] D. Zeinalipour-Yazti, "Exploiting the security weaknesses of the gnutella protocol", Technical Report CS260-2, Department of Computer Science, University of California, 2001.

[30] N. Anderson, "Hacking Digital Rights Management", *Ars Technica*, 2007, http://arstechnica.com/apple/news/2006/07/drmhacks.ars.

[31] MPEG-21 Consortium, "ISO/IEC FDIS 21000-17:2006(E) MPEG-21 - Part 17: Fragment Identification of MPEG Resources", ISO, 2006.

[32] MPEG-21 Consortium, "ISO/IEC FDIS 21000-15:2006(E) MPEG-21 - Part 15: Event Reporting", ISO, 2006.

[33] MPEG-21 Consortium, "ISO/IEC FDIS 21000-2:2005(E) MPEG-21 - Part 2: Digital Item Declaration", ISO, 2005.

Author Biographies

H. Castro, 5 year-long engineering degree in Electrical and Computing Enginery, awarded by University of Porto, Porto, Portugal, 2003. multimedia content distribution, rich media content structuring; p2p networking.

A. P. Alves, is a full professor at the University of Porto where he is teaching in the area of Telecommunications. His research activities have been at INESC Porto, a research institute that he founded in 1985 to work in the interface between the academia and industry. He has been a member of INESC Porto direction since then until April 09. At INESC Porto he was responsible for research in telecommunications and, in particular, in digital television technologies, looking at networked file-based production systems. In this area he was responsible for a long term program of collaboration with BBC R&D. He obtained his PhD from University of Bradford in 1981 has been actively involved in national and European projects. He supervised many MSc and PhD students in the area of telecommunications and multimedia. He was also involved in the creation of several companies span off INESC Porto.

M. T. Andrade, PhD in Telecommunications, Electrical and Computing Engineering, awarded by the University of Porto, Porto, Portugal 2008. multimedia networking, context-aware multimedia, content adaptation.