Music2Share—Copyright-Compliant Music Sharing in P2P Systems

TON KALKER, FELLOW, IEEE, DICK H. J. EPEMA, PIETER H. HARTEL, R. (INALD) L. LAGENDIJK, AND MAARTEN VAN STEEN

Invited Paper

Peer-to-peer (P2P) networks are generally considered to be free havens for pirated content, in particular with respect to music. We describe a solution for the problem of copyright infringement in P2P networks for music sharing. In particular, we propose a P2P protocol that integrates the functions of identification, tracking, and sharing of music with those of licensing, monitoring, and payment. This highly decentralized music-aware P2P protocol will allow access to large amounts of music of guaranteed quality; it merges in a natural way the policing functions for copyright protection and an efficient music-management infrastructure for the benefit of the user.

Keywords*—*Content-aware networks, digital rights management (DRM), fingerprinting, multimedia, music sharing, peer-to-peer (P2P), search protocols, watermarking.

I. Introduction

While the music-recording industry is struggling to fight music-sharing technology such as provided by KaZaa [1] with legal as well as technological means [2], huge numbers of Internet users are turning to music-sharing applications based on peer-to-peer (P2P) technology [3]. However, the concept of *legal* music sharing embedded in an acceptably *secure* P2P framework has barely been put to the test. In this paper we propose an architecture called Music2Share (M2S) in which secure content sharing and P2P networking coexist.

A number of explanations have been offered for the success of (illegal) music sharing over the Internet. First, it is supposedly due to the for-free access to digitally perfectly copied yet pirated content, the underlying argument being that many consumers feel that the price of prepressed music is too high. Second, the choice of music on the major P2P networks is almost unlimited, certainly in comparison to what is offered by regular retail shops. Third, there is a trend that consumers are no longer interested in complete CD albums, but only in particular tracks. Why would users buy CDs if P2P networks allow them to collect any tracks of their interest and compile their own CDs? Fourth, the number of households with fast connections to the Internet, which makes downloading (and uploading) music more convenient, is increasing rapidly.

Consequently, digital networks in general and P2P networks in particular are indeed currently operating as free havens for pirated content, in particular, music. However, the current success of illegal music distribution over the Internet does in itself not provide evidence that commercial, copyright-compliant online music selling and sharing is and will remain unfeasible. First evidence is provided by *iTunes* [4] from Apple, a central-server based system that offers a relevant collection of music for the price (at the time of writing) of US$0.99 per track. The popularity of iTunes shows that users are willing to pay for content if the online music service is sufficiently compelling (with respect to music quality, ease of use, and availability of relevant music).

Central-server based systems for electronic music delivery have the distinct disadvantage of a bandwidth bottleneck at the central server(s). In such systems, two users who live in close proximity (in cyberspace) and are buying the same track still need to download from this central server, whereas they could more easily have shared the same track via a local

Manuscript received September 7, 2003; revised December 28, 2003.

T. Kalker is with Philips Research Eindhoven, Eindhoven 5656 AA, The Netherlands, and also with the Technical University Eindhoven, Eindhoven 5612 AZ, The Netherlands (e-mail: ton.kalker@ieee.org).

D. H. J. Epema is with the Parallel and Distributed Systems Group, Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, Delft 2600 GA, The Netherlands (e-mail: D.H.J.Epema@ewi.tudelft.nl).

P. H. Hartel is with the Distributed and Embedded Systems Group, Department of Computer Science, University of Twente, Enschede 7500 AE, The Netherlands (e-mail: pieter.hartel@utwente.nl).

R. I. Lagendijk is with the Information and Communication Theory Group, Faculty of Electrical Engineering, Mathematics, and Computer Science (EEMCS), Delft University of Technology, Delft 2628 CD, The Netherlands (e-mail: R.L.Lagendijk@ewi.tudelft.nl).

M. van Steen is with the Department of Computer Science, Vrije Universiteit, Amsterdam 1081 HV, The Netherlands (e-mail: steen@cs.vu.nl).

Digital Object Identifier 10.1109/JPROC.2004.827368

0018-9219/04$20.00 © 2004 IEEE

connection on a P2P network. From the viewpoint of the efficient use of storage and bandwidth, an online music service is better organized as a P2P network. This observation has been put in practice by Altnet [5], which operates as a sub-P2P network under KaZaa [1].[1]

An important motivation for our work is the firm belief that it is worthwhile and challenging from both a technical and an economical perspective to develop technologies that enable and stimulate legal music sharing over the Internet. Such technologies will only provide a viable solution for copyright owners if hooks are present for enforcing copyright compliance and for guaranteeing revenue, while for users they will only be an acceptable replacement if quality, low costs, and a large collection of tracks are guaranteed.

The paper is organized as follows. In Section II we discuss the points of departure for our M2S architecture. Section III gives an overview of the M2S architecture, while Section IV provides an initial analysis of its feasibility. Related work is discussed in Section V, and we present our conclusions in Section VI.

II. Points of Departure

In this section we present the points of departure for the design of the M2S architecture and the problems that M2S addresses. We start by observing that decentralized networks are in our opinion the best architecture for content distribution and that audio fingerprinting may be used as a tool to provide persistent identification. These observations are made from a bird's eye perspective, but after a more careful consideration, it appears readily that many problems need to be solved before legal sharing of music over open (P2P) networks is viable.

A. Fingerprinting

A little contemplation will reveal that an essential ingredient in creating a secure music-sharing network is the ability to establish the (perceptual) identity of audio files.[2] In the case of Napster, this ability was tried using text-based methods (file names, ID3 tags in MP3 files, etc). However, because such textual information can easily be modified by ordinary users, this strategy turned out to be not very successful in establishing secure identification.

The solution proposed in this paper is to deploy a more robust audio identification technology known as *audio fingerprinting*. In analogy with human fingerprints, audio fingerprints provide accurate and compact descriptions of (segments of) audio. Such fingerprints are often based on psychoperceptual properties by representing the perceptually most relevant aspects of music. In M2S, fingerprinting is used to identify and subsequently replace low-quality files with high-quality ones. In this way, we ensure that users always have access to music of the quality they are entitled to. Audio fingerprinting technologies are currently being offered by several companies such as Audible Magic [6], Relatable [7], Shazam [8], and Philips [9].

B. Decentralization

An important second point of departure of this paper is that we seek fully decentralized solutions, both for storing (encrypted) music files and, in contrast to current approaches, for storing and accessing fingerprints. Such solutions solve two problems inherent to the currently applied client–server architectures. The first problem is that a client has to reference explicitly a host in order to use its service, which hinders content-based searching, as it essentially requires a client to search each online music vendor separately. Aggregate search engines are possible, but face considerable difficulties due to copy protection restrictions and privacy issues. The second problem is that the maintenance of and load balancing across centralized servers is costly and complicated, the more so when considering that soon a billion users may be both producing, archiving, and consuming content. The concept of centralized and dedicated content servers is becoming less natural as a storage architecture in such an environment.

A promising and rapidly emerging approach to solving the above-mentioned problems is to organize servers into a P2P network in which the nodes maintain their independence while providing the facilities for efficiently routing search requests to the appropriate nodes. To some extent, also organizational and logistic problems are alleviated by the inherent fault tolerance of P2P systems. The simplicity of the P2P protocols has the additional advantage that any type of node can participate in a P2P network, whether it is a low-end personal computer or a high-end server. Moreover, nodes are allowed to join and leave at will without seriously disrupting the overall performance of the system. The P2P approach has already been successfully applied to building large-scale distributed storage systems such as CFS [10], Past [11], and OceanStore [12].

C. Problems

Despite the advantages of P2P systems mentioned above, there are also two major problems that P2P systems need to solve before they are suitable for legal music distribution. First, current P2P systems do not support efficient content-based searching, i.e., searching using an intrinsic and inalienable attribute of the content (e.g., a fingerprint or certified metainformation) rather than using the name of an artist or the title of a song. So-called structured P2P systems such as those mentioned above can only operate efficiently if data are explicitly identified. In contrast, unstructured P2P systems such as Gnutella [13] do offer facilities for content-based searching but at the price of a (much) lower performance.

Second, current P2P systems lack security: they do not offer payment, protection against unauthorized access, guaranteed quality, etc. Only recently research has started on building secure P2P networks (see, e.g., Castro *et al.* [14] or Grimm and Nützel [15]). Initial attempts at commercial deployment of secure and anonymous P2P systems is being tried by a small number of initiatives such as Earth Station Five [16]. No system is available yet that guarantees the

[1] It is too early to tell whether or not Altnet is successful in its business model; public figures are not available at the time of writing.

[2] An audio track in, say, wave format or MP3 format is perceptually the same, of course assuming proper encoding; the format is ideally transparent to both the user and the copyright owner.

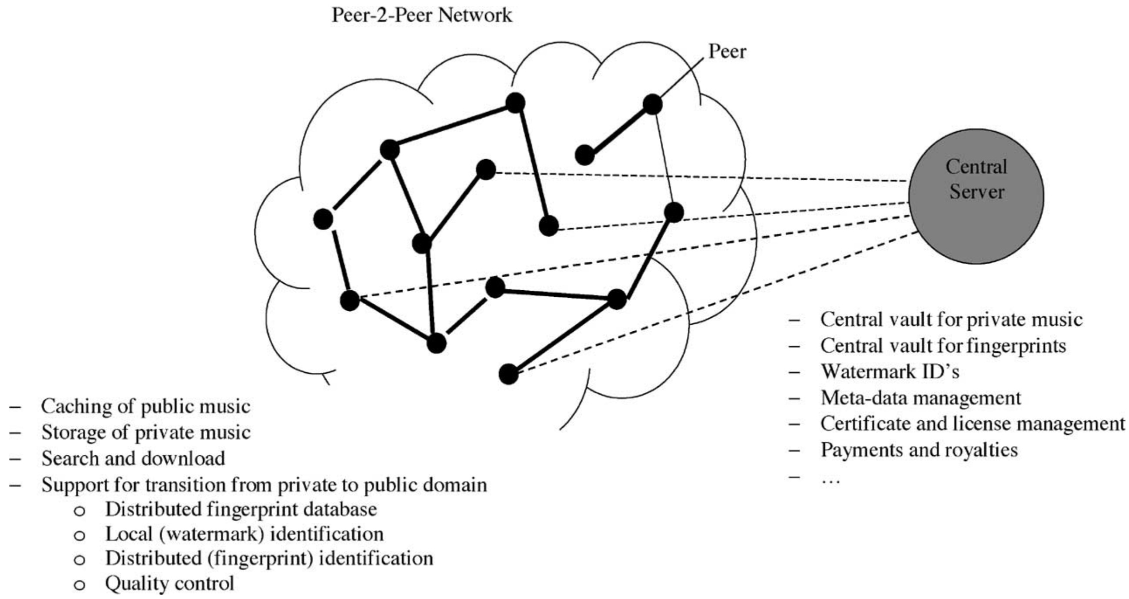
Peer-2-Peer Network
Peer
Central Server
– Central vault for private music
– Central vault for fingerprints
– Watermark ID's
– Meta-data management
– Certificate and license management
– Payments and royalties
– …
– Caching of public music
– Storage of private music
– Search and download
– Support for transition from private to public domain
o Distributed fingerprint database
o Local (watermark) identification
o Distributed (fingerprint) identification
o Quality control

Fig. 1. Overview of Music2Share architecture.

wishes of both the copyright owners and the consumers of music.

We argue that both problems can be solved elegantly by M2S. (The reader should be aware that to date there exists no operating M2S network.) This paper aims to present an architecture for copyright-compliant music sharing based upon P2P protocols, cryptographic algorithms, watermarking methods, and perceptual-fingerprint-based music identification.

The basic premise of M2S deviates from the mainstream ideas on legal music sharing, which focus on locking up content and enforcing digital rights management (DRM) rules by cryptographic means. Below we will argue that there is another viable approach that only charges the trading of music, but leaves private use of music free of restrictions. The inherent quality control of M2S, the lack of *technical* usage restrictions, and the easy payment structure make it an interesting alternative to many existing propositions for music sharing on P2P networks.

III. M2S Architecture

The basic philosophy of the M2S architecture (see Fig. 1) is that the origin of an audio file is of no relevance. If a user of the M2S network locates an audio file in which he is interested, he is allowed to download it provided that he somehow pays for it. It does not matter whether the file was originally obtained from another online music service (e.g., iTunes), ripped from a CD by the administrator of the uploading host, or obtained from another peer in the M2S network or any other P2P network. The only relevant fact is that the user will obtain a copy of the requested audio file and that appropriate royalties must be paid.

To understand the overall M2S architecture and to appreciate how it provides secure yet decentralized content management, we discuss in this section the basic elements of content sharing, payment, content identification, and audio file upgrading as supported by M2S.

A. Content Management

In the M2S system, we distinguish three types of content. First, there is *public content*, which consists of encrypted audio files which are distributed and replicated across the machines of the M2S P2P network, effectively forming a large distributed *public database* of encrypted audio content. In general, it can be expected that this public content is directly provided by music distributors.

Second, each user has his own *private content* consisting of unencrypted audio files residing at his own machine that have been obtained in a legal way, that cannot be shared with other users, and that constitute a user's *private database*. This content is available to the buyer–user, who can do anything with it he likes: rendering, copying, processing, etc. A user can fill his private database in various ways. He can extract files from the public database to be incorporated into his own private database provided he is authorized to do so. There are various ways in which this authorization can take place, as we discuss below.

Another way for a user to obtain private content is by buying a CD and uploading tracks from it to his private database. In return, the user receives a *token* by which he can prove that he is authorized to extract that same content from the public database. For example, a user wanting to download an audio file to his MP3 player only needs to retrieve the appropriate token and pass it to the M2S system to extract files from the public database. A convenient way of storing tokens is by means of a simple smart card. Such a mechanism will also allow users to *buy* tokens. Content that has been legally obtained by extraction from the public database or by other means is also referred to as authorized content.

The third type of content in M2S is *nonauthorized content*, which consists of audio files on users' machines that cannot be reliably authenticated (for example, badly compressed files downloaded from other P2P networks or recordings from your own for-fun garage band). M2S strives to identify and authenticate every audio file on its network by external (certificates) or internal (watermarking) labeling or by recognition (fingerprinting). If successful, these files enter into the authorized domain. If not, M2S does not take special measures to prevent nonauthorized content from spreading across the network. First, because we propose that the M2S network behaves as much as possible as a normal P2P network with the default assumption that in general users are honest. Second, as we explain below, we believe that there will be so little unauthorized traffic that taking special measures against it does not warrant the effort.

Files that are stored in private databases are automatically copied, encrypted, and subsequently stored by the M2S system in the public database. When and where private content is transferred to the public domain is completely outside the control of the users. Likewise, where and how public content is distributed, replicated, and physically stored is also completely transparent to end users. It is the task of the M2S system to ensure that the public content is managed efficiently, guaranteeing optimal performance of searching by users. From the point of view of the user, the M2S system behaves as a user-friendly interface to a large variety of high quality content. It allows him to mingle in a completely transparant way music bought in regular retail shops as well as music bought online. The M2S system is responsible for the management of all musical content, including identification, authentication, encryption, and distribution.

B. Royalty Payment

An issue in the payment scheme of a music-sharing network is whether users should be charged for downloading files or for playing them. The latter approach has been taken by Altnet [5], which actually encourages users to download copyrighted files onto their hard disks to achieve more efficient content distribution. Users are even given bonus points (*peer points*) that will allow them to obtain licenses to buy content. This approach has the obvious advantage that the copyright of local copies is not an issue, thereby avoiding the difficult copyright issue of what actually constitutes a local copy or cache copy. The disadvantage of the Altnet method is that audio files need to be encrypted (in order to bind the license to the audio file) and that it will be difficult for a user to use the content on any other device than his PC, for example, on a portable MP3 player. As long as there is no common and easy-to-use DRM standard for portable audio players, the broad acceptance of encrypted audio files will be a difficult issue.

The position taken by M2S is that obtaining and using audio content should be at least as easy as buying a CD. Once purchased, the user should be able to use the audio for private use as he sees fit: burning backup copies on CD, downloading to portable players, listening on any device of his choice. Moreover, using simple digital-to-analog-to-digital (DAD) conversions, it is always possible to remove any encryption layer from a protected audio file. However, *unauthorized spreading of the content should be prohibited as much as possible* and this is where the M2S architecture steps in by identifying, tracking, and filtering audio content as it flows over the M2S network. However, the idea of offering incentives to users of the M2S network for making their computers and connections available for storing and distributing private content is certainly considered an option within M2S.

C. Content Authentication

The proposed M2S network consists of a classical P2P network enhanced with a central trusted party (TP). The P2P part of M2S implements the public database containing encrypted audio files. This part also assists the TP with establishing the identity and the quality of audio files in the private databases. The TP is responsible for authenticating audio files based upon their identification, and for attaching digital certificates to them. These certificates constitute the essential hook in searching and in assuring payment of royalties when audio content is transferred from the public database to a user's private database. The M2S network, therefore, needs to establish the identity of audio files, and link this identity to a license system with an appropriate payment infrastructure. When an audio file enters the M2S network, its identity is not necessarily easily obtainable as a metadata field. We distinguish three methods for the identification of audio files.

1) Identification by Authorized Upload: In case an audio file is provided by an authorized server, it may have a digital certificate associated to it which securely identifies the file (as well as its quality, and license and copyright information). Locating and exchanging such a file and the associated certificate can be done with common P2P protocols. In M2S, the audio files in a user's private database are not encrypted, and the personal usage of such a file is not restricted in any way. If an audio file is modified (e.g., by compressing or cropping it), the associated certificate is no longer valid, and the M2S network will not necessarily transfer the file to the public database (and thus make it available to other users) as a public file.

2) Identification by Watermark: It is possible to transform authenticated content into nonauthenticated content, for instance, by compressing or transcoding it, or by analog-to-digital conversion followed by reencoding. The identity of the transformed content may be difficult to establish (metadata fields are typically lost). A popular solution to this problem is to embed the identity of the song (and possibly other licensing information) with a digital watermark (typically as a barcode-like number). Alternatively, audio files may enter the M2S network without the intervention of an authorized server. The M2S network may try to establish the identity of such a file by checking whether or not an M2S watermark is present. If such a watermark is found, the network is able to retrieve proper licensing information from an authorized central server. Issues to be resolved when employing watermarking are where to do the watermark embedding and where to do the watermark detection.

It is well recognized that embedding high-quality watermarks is a delicate issue in terms of complexity, security, and quality. It seems natural to entrust the authorized central servers or the music producers with this task. This offline approach allows the use of ample computational resources and expert human quality control. In fact, embedding a watermark is probably best treated as a part of the content creation process. Then, any content originating from an authorized server or ripped from a CD is easily identifiable on the M2S network. In case the watermark is inserted by the authorized server, it will probably not be economically feasible to use human intervention for quality control and the watermark insertion will have to be fully automatic. Note however, that the M2S watermark is only for audio identification purposes, not for personalization purposes (as in forensic tracking). The embedding process can, therefore, be done offline, allowing ample resources (computational, time, storage, and otherwise).

When an audio file without a certificate is added to the private database of a user, the lack of a certificate is easily assessed by the M2S client. Before such an audio file can be shared as a public file, the client will need to assess its identity and quality. As reading a watermark is a cheap operation, it can be performed by the client. If a watermark is successfully read, the information in the watermark (a small number of bytes) is submitted to the M2S TP to obtain a proper certificate to be attached to the audio file. At this point the audio file is identified, but this is not sufficient for allowing it into the M2S public database: what is still missing is an assessment of its quality.

3) Quality Control: Quality control is important for user acceptance of the M2S architecture, and means to establish quality are, therefore, essential. There are several options to do so. First, semifragile watermarks can be used which are robust in the face of mild degradations but will become unreadable with more severe degradations. With such watermarks, the mere detectability of a watermark is a sign of sufficient quality. A second option is to do explicit quality control, which can be performed by an authorized server or by the clients. The former possibility may not be optimal because of the large amount of resources needed at the server, so, in line with the M2S philosophy, quality control is best performed by the clients. In order to make this possible, the authorized server sends sufficient perceptual data to the client to allow it to do quality analysis.

4) Identification by Fingerprint: In the above we assumed that song identification can be done locally at the clients, by reading either a certificate or a watermark. However, it will be a long time before all songs have a watermark, if ever. Therefore, the M2S architecture will have to identify audio files that have neither a certificate nor a watermark. This is where audio fingerprinting enters the scene. A song in a user's private database with no explicit identification information can still be identified by extracting a fingerprint from it and querying a database of fingerprints. The extraction of a fingerprint is typically a cheap operation that can be done by an M2S client. Unfortunately, typical querying solutions are centralized and, therefore, do not match the M2S philosophy. Fingerprint searching is a fuzzy process that requires considerable computational resources, and fast fingerprint searching is only possible if it can be parallelized across multiple partitions of a distributed fingerprint database. M2S takes this approach to the limit by spreading the fingerprint database over all M2S clients. How to build an efficient distributed P2P fingerprint search algorithm is still part of ongoing research.

D. Audio File Upgrading

It is possible that the M2S identification and quality tools have verified that a user is in the legal possession of some music file, that this file is not of the best quality, and that an equivalent quality-assured version exists at an authorized server. Then, M2S will automatically transfer the quality-assured file from the public database to the user's private database. That is, in the philosophy of M2S, the possession of a version of a song of sufficient quality entitles the user to a quality-assured version of the song with all the associated rights of playing and copying. Of course, this approach begs for abuse of the M2S system. What mechanism will stop a user from (illegally) obtaining a bad-quality version of a song and using the upgrade mechanism of M2S to obtain a good-quality version of it? There are several answers to this question.

First note that this kind of illegal trading on the M2S network itself is extremely difficult on a large scale. Large-scale trading requires efficient and public search protocols. As M2S is a controlled P2P network, this kind of large-scale trading is easily stopped before it ever takes off. Small-scale trading will exist, but is difficult to prevent and probably has as much effect as trading of files using e-mail: it can be done, but is extremely cumbersome. Large-scale illegal trading on other type of networks (KaZaa, Gnutella) cannot be prevented by this approach directly other than by legal action.

Indirectly, M2S provides strong incentives to abandon other P2P networks in favor of M2S: 1) guaranteed quality; 2) automatic upgrades of songs; 3) uniform and reliable metadata; and 4) associated music organizer tools. Given the general human desire to keep things simple, it is not expected that the general public will use two file-sharing networks, one for illegally obtaining (bad-quality) songs and one for automatic upgrades. Extending the M2S philosophy, the possession of CD tracks will allow an M2S user to obtain quality-controlled compressed versions over the P2P network as an alternative to private ripping.

IV. Discussion and Analysis

In this section we provide a discussion and a preliminary analysis of some of the most salient features of our proposed architecture for secure content sharing. We split the discussion into three parts, namely, about issues related to P2P networks, to coding, and to protocol security issues.

A. P2P Analysis

P2P systems in general, and P2P systems for music sharing in particular, are extremely popular. For instance, KaZaa claims at the time of writing that their software has been downloaded upwards of 230 million times. Measurements on the U.S. Internet backbone have shown that the fraction of traffic due to Gnutella in 2001 was about 1.2%—a seemingly low percentage, but very high considering that Gnutella was at the time only about two years old. One can only conclude that the efficiency of any proposed P2P protocol is of paramount importance.

As the aim of M2S is to give its users guarantees that if a music file exists it is found, using or modifying *classical* P2P protocols like Gnutella and Freenet is not an option, because these only give probabilistic guarantees of finding an existing file. Therefore, M2S will need to consider the *second-generation* deterministic, structured overlay networks that are based on distributed hash tables (DHTs) (see Section V).

As far as performance is concerned, in comparison to existing P2P music-sharing protocols, M2S deviates in two ways. First, there is an additional protocol step for security: retrieving a decryption key once an audio file has been located in the public database. This requires only two additional small messages per retrieval, which, relatively speaking, does not add much to the network load. As the decryption is performed on the users' machines, we do not include that in the system load.

Second, the M2S architecture aims to bring private content back into the public domain. For authorized content, this only brings minimal overhead, as identification by watermark retrieval and quality analysis can be done on the local client. However, for nonauthorized content, identification is only possible using audio fingerprints. This requires a distributed implementation of a fuzzy fingerprint search engine. Important questions that still need to be solved are the (dynamic) distribution of the fingerprint database over the peers of the network, protocols for distributing fingerprint search requests over the network, and mechanisms for merging identification messages in case of several and possibly conflicting identification answers. An important aspect in the proposed distributed search mechanism is that audio fingerprints are typically much larger than purely cryptographical fingerprints. This implies that storing and transferring audio fingerprints is not necessarily an insignificant part of the traffic on the M2S network. Initial experiments have shown that without proper care, identification traffic can easily clog up the complete P2P network.

B. Coding Analysis

Watermarking and fingerprinting have been recognized as valuable tools for content recognition. The inclusion of these two technologies in the M2S architecture, however, poses some particular issues and challenges.

In most copyright protection applications, the robustness of an embedded watermark is of prime importance. The loss of a watermark (i.e., the inability by a watermark reader to detect a watermark) usually means that the content is no longer protected. This is not the case in M2S: music is in this case still protected but the burden of identification is pushed to the most complex level, namely, identification by audio fingerprinting. The impact is, therefore, more upon performance than upon functionality. Consequently, the M2S watermark can be designed with less emphasis on robustness than is usual and therefore, with more emphasis on inaudibility and security.

With respect to the latter, the most relevant attack is the *copy attack* [17]. The aim of such an attack is the unauthorized insertion of a watermark, whereby watermark secrets are obtained by estimation. If this type of attack is successful on a large scale, it would mean a serious compromise of the identification functionality of M2S. Note, however, that due to the use of quality-checking tools in M2S (see below), it is easy to identify spoofed watermarks. Therefore, the main worry is not so much misidentification as well as nonidentification. A solution to this problem is the use of *content-dependent watermarks* as, for example, proposed in [18]. An attractive property of the solution proposed by [18] is the use of audio fingerprints for binding watermarks to the audio content: as audio fingerprints are an essential part of the M2S architecture, the overhead created by making watermarks more secure against copy attacks need not be excessive. The precise details of such a solution, as well as other security issues, are still a topic of research.

A new challenge for M2S that is not very common in the academic literature is reliable, lightweight, and automated quality control of audio files. Once a file has been identified, either through watermark detection or fingerprint search, the quality of the file has to be estimated before it can participate in the file upgrade protocol. In the most extreme case, quality control must guarantee that the watermark has not been spoofed. In more subtle cases, the quality tool, for example, needs to reliably estimate whether or not an audio file corresponds in quality to 32 kb/s MP3 or to 192 kb/s MP3. The result of such an estimate may determine how much a user has to pay for un upgrade. A number of approaches to this problem can be taken. First, for authorized content, the degradation of the embedded watermark may be taken as a rough quality tool. Second, for all content, the error rate in fingerprint matching may serve as an indicator of quality. However, it is to be expected that without special measures, both watermarks and fingerprints will be unreliable quality indicators. The design of watermarks and fingerprints that can act as quality indicators is an active topic of research for M2S. Of course, other quality control methods may also be envisioned. For example, next to searchable fingerprints, the central server may compute and distribute *special fingerprints* that have limited search capabilities (or none at all), but may assist in determining audio quality.

C. Protocol Analysis

We will now present the overall protocol of M2S. In the discussion below, the numbers refer to the components, messages, and computations in Fig. 2.

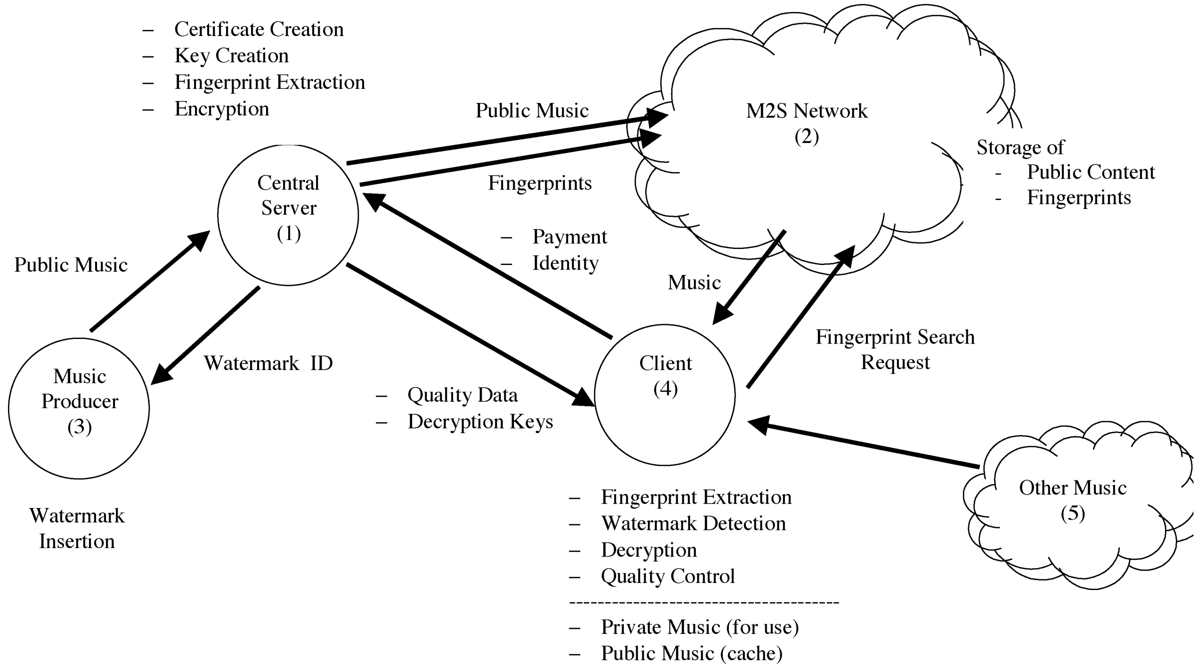

Fig. 2. Music2Share protocol overview.

The M2S architecture consists of a set of central authorized servers [(1) in Fig. 2], of the M2S P2P network (2) that is seeded from these servers, of music producers (3) who upload music to these servers, and of clients (4) who download music from the M2S P2P network on behalf of users. The public music files on the central servers are in encrypted form, and for each such file there are associated certificates, watermarks, fingerprints, and a decryption key. The private music on the users' disks is in the clear (plaintext) and is either directly derived from public music by decryption or obtained from other sources (e.g., ripping from a CD) (5).

The trusted computing base is small: we assume that the music producers and the server(s) form a secure domain, that the client is a secure application (on users' machines), that the payment devices on the users' machines are secure, and that the communication between the clients and the server(s) is secure. We make no security assumptions about the peers or the users. The peers and the traffic to and from the peers is encrypted by the protocol and may thus be transported freely on an open network. The music received by the user is potentially watermarked with the identity of the client for forensic tracking purposes (we do not pursue this matter any further in this paper).

We now present four scenarios of the use of M2S, one for uploading and three for downloading music. In the latter three scenarios, which are of increasing computational complexity, the user gets the music he wants, with guaranteed quality, except when he cannot or refuses to pay.

1) Scenario 1: Upload: A music producer chooses some music, negotiates a watermark ID with the server, embeds the watermark into the music, and uploads it onto the server. The server receives the music, and calculates a certificate (incorporating relevant metadata) that will identify this authorized music uniquely. Also, an audio fingerprint is calculated for the purpose of identifying nonauthorized music. The server then chooses an encryption key and encrypts the music. The key is stored with the certificate and the fingerprint. Appropriate peers store the encrypted music and certificates for future reference. Other peers (which may be the same or different) store (parts of) the fingerprints and pointers to the associated certificates on the central server.

2) Scenario 2a: Explicit Download: A user requests some music from a client by metadata (explicit request), by a watermark ID (implicit watermark request), or by a fingerprint (implicit fingerprint request). We assume that the client receives a valid token from a smart card or some other secure payment device. The client then asks the server for the key corresponding to the requested audio file. The client also receives the encrypted music from the P2P network; the music can now be decrypted. If forensic tracking is enabled, the music is also watermarked with the identity of the client. The result is sent back to the user.

3) Scenario 2b: Watermark Request Download: The user places some music without a certificate on his disk that is derived from private content. The client notices the lack of a certificate and reads the watermark ID from the music file. The retrieved ID is sent to the central server for retrieval of quality-checking data. Depending on the estimated quality of the music file and the requested quality by the user, a payment token is exchanged with the server, and the original user file is replaced by a certified music file from the M2S network in the same way as in Scenario 2a.

4) Scenario 2c: Fingerprint Request Download: The user places some music without a certificate on his disk that is *not* derived from private content. The client notices the lack of a certificate and tries to read the watermark ID from the music file. After failure to do so, the client computes fingerprints from the music file and sends a request for identification to the network. If successful, the client sends the retrieved ID to the central server for retrieval of quality-checking data. The rest of this scenario proceeds as Scenario 2b.

V. Related Work

A. P2P Systems

The last few years have seen a tremendous interest and development in P2P systems, whether for music sharing or for other purposes. The two most widely used P2P systems for music sharing are Napster [19] and KaZaa [1]. Napster, which had to discontinue its operations for legal reasons, employed a centralized architecture for name-based lookup; only after a user had obtained the location of the desired file, the actual file sharing was done in a true P2P fashion. As of this writing, KaZaa provides probably the most popular music-sharing program, but little is publicly known about it except that the architecture has a heterogeneous structure.

Two other early P2P systems are Gnutella [13] and Freenet [20]. Gnutella is query based in that it looks for potentially multiple matches to a request, uses a broadcast-based search algorithm, and replies with the IDs of the nodes with the desired contents. Freenet is file based in that it looks for specific files, uses a single search path by sending the request for a file in each hop to the node for which the local cache indicates the presence of a file that is "close" (in terms of file identifiers), and responds by sending the file in the reverse direction along the search path. In order to bound the searches, both Gnutella and Freenet employ time-to-live counters in their search requests and do not guarantee finding the required content even if it exists in the network.

The operating systems research community has developed P2P protocols like Chord/CFS [21], [22], Pastry/PAST [11], [23], and Tapestry [24]. In these systems, generally referred to as being based on DHTs, both nodes and files have identifiers (of size 128–160 b) derived from applying hash functions to some of their characteristics, such as IP numbers or keywords. In Chord, the node IDs are arranged in a virtual ring directed according to the binary values of their IDs, and the responsibility for a file rests with the first node after its ID in the ring. Other DHT-based systems deploy a similar scheme. This immediately suggest an albeit naive way with deterministic guarantees for finding content: simply go along the virtual ring, one step at a time, until the file is found or known not to be present. To speed up this algorithm, DHT-based systems have nodes maintain a table with the IDs of the nodes that are at distances about equal to powers of two, speeding up the searches to be logarithmic in the number of potential nodes.

Unlike M2S, all of the above protocols are oblivious to the kind of content distributed.

B. Watermarking

Watermarking is the art of imperceptibly hiding information into multimedia content [25]. In its robust form, it can be used to signal copy protection states (e.g., "copy never") or embed personalization information (e.g., identifying the buyer of a song for *forensic tracking* purposes). In its fragile form, it can be used to signal content modifications. The main strength of a watermark is that it *can go where no encryption solution can go*, namely, to the clear-content domain (in particular, the analog domain). As such, when properly applied, it can serve to enhance and/or protect management and delivery of (music) content. Although the concept of watermarking is already quite old, it has only resurfaced with full strength in electronic form since 1996. A number of efforts have been undertaken to apply watermarking to content protection, but none of these initiatives have been taken to full fruition, the most infamous example being the Secure Digital Music Initiative [26]. However, there is still a strong belief in the market that watermarking will find a successful application.

Fingerprinting is the art of creating perceptual summaries. In analogy with cryptographic hashes, fingerprints are bit strings that are much smaller than the original multimedia objects, but still sufficient to identify that object [9]. In contrast with cryptographic hash functions, fingerprints depend in a continuous manner on the multimedia object whereas cryptographic hashes are bit sensitive. For fingerprints, however, small perceptual changes in the original object should result in small fingerprint changes. Fingerprints are useful in automatic content recognition (e.g., in forensic tracking applications) and as a tool for added watermark security. Several business initiatives such as Shazam [8] and Audible Magic [6] reflect the potential of audio fingerprints both for DRM and enhanced music experience.

C. DRM

Digital rights need to be encoded in some language, and for these languages to be machine-interpretable, they must have a well-defined syntax and a proper semantics. Several rights expression languages have been developed, such as Digital Property Rights Language (DPRL), eXtensible rights Markup Language (XrML) [27], and Open Digital Rights Language (ODRL) [28]. They provide a rich syntax and structure that allows fine-grained specification of control over digital contents. These languages are able to express different kinds of rights and a myriad of terms and conditions, but their interpretation relies solely on human intuition. Recent work points in the direction of logics for licenses [29], [30].

There are several DRM platforms on the market: SealedMedia Enterprise License Server (ELS [31]), Microsoft Windows Media Technologies [32], IBM Electronic Media Management System (EMMS [33]), Sony's OpenMG [34], InterTrust Rights Systems [35], and ContentGuard RightsEdge [36].

Consumer electronic devices with DRM functionality are a relatively new trend in the content industry. Eskicioglu and Delp [37] provide an overview of content protection on consumer electronic devices such as set-top boxes, TVs, VCRs, and DVD players.

Grimm and Nützel [15] propose a combination of DRM with P2P file sharing based on the idea that a user who has paid for content might like to earn revenue through redistribution. Nonpaying users are prohibited from earning money through redistribution, thus providing a strong incentive to pay for content. The *earn if you pay* model is complementary to the *quality guaranteed if you pay* model that we explore in the M2S architecture.

The concept of Light Weight DRM (LWDRM [38]) has been introduced by the Fraunhofer Institute. LWDRM embeds the identity of the user who downloaded the content in a watermark. The watermarked content can be used freely for all legal purposes, particularly *fair use* (see [39, Sec. 107, Tit. 17, Ch. 1], Fair Use Doctrine: *fair use of copyrighted content, including reproduction for purposes such as criticism, comment, news reporting, teaching, scholarship, or research does not violate or infringe the copyrights*). However, when the user engages in illegal activities, the embedded watermark can be used to trace back to the fraudulent user. M2S provides benefits to honest users (such as guaranteed quality) and is not primarily aimed at catching users who cheat. M2S uses watermarks to identify content and not to identify users, as LWDRM. In a sense, the aims and means of M2S and LWDRM are orthogonal. A system that combines M2S and LWDRM technology would also be feasible.

Feigenbaum *et al.* [40] warn of the risk of privacy infringement caused by DRM systems. We use anonymous payment and anonymous downloads in the P2P network to maintain anonymity of the user. The anonymity of the user will be violated once she starts sharing (watermarked) content she has paid for. Paradoxically, she may and should share content that she has not paid for—encrypted content.

VI. Conclusion

In this paper, a novel approach to music sharing on P2P networks has been sketched that will both satisfy the user as well as the content owner. From the viewpoint of the user, M2S will offer a music-sharing network with no technical restrictions on content that has been bought. Moreover, the M2S network will assist the user in managing (attaching proper metadata) and upgrading (with controlled quality) of his own private content. From the viewpoint of the content owner, the M2S approach offers an efficient music-distribution mechanism, exploiting the computational, bandwidth, and storage resources available on the Internet. Equally important, all music sharing on the network is controlled and payments are guaranteed for all music trading. The M2S architecture consists of a P2P network and a central authority. The former is responsible for storing, transferring, identifying, and controlling the quality of the music files on the network. The latter controls the P2P network and is in particular responsible for all integrity checking and payments.

The basic technologies for the proposed architecture are currently available (P2P technology, watermarking, fingerprinting). However, the application of these technologies in the proposed music-sharing architecture still has to be worked out and refined. This holds in particular for the next generation of P2P protocols, the identification and quality-checking tools, the security issues, and the payment protocols. Progress on these topics will be reported in future publications.

References

[1] KaZaa [Online]. Available: http://www.kazaa.com

[2] B. May and M. Singer, "Unchained melody—the digitization of music has industry execs in a twist," *McKinsey Q.*, vol. 2001, no. 1, pp. 128–137, 2001.

[3] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, "An analysis of Internet content delivery systems," *ACM SIGOPS Oper. Syst. Rev. (Special Issue on Network Behavior)*, vol. 36, pp. 315–327, Winter 2002.

[4] iTunes [Online]. Available: http://www.apple.com/itunes

[5] Altnet [Online]. Available: http://www.altnet.com

[6] Audible Magic [Online]. Available: http://www.audiblemagic.com

[7] Relatable [Online]. Available: http://www.relatable.com

[8] Shazam [Online]. Available: http://www.shazam.com

[9] J. A. Haitsma and T. Kalker. (2002, Oct) A highly robust audio fingerprinting system. *Proc. 3rd Int. Conf. on Music Information Retrieval (ISMIR)* [Online]. Available: http://www.research.philips.com/Assets/Downloadablefile/ismir2002Fingerprint-1533.pdf

[10] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide-area cooperative storage with CFS," in *Proc. 18th Symp. Operating System Principles*, 2001, pp. 202–215.

[11] A. Rowstron and P. Druschel, "Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility," in *Proc. 18th Symp. Operating System Principles*, 2001, pp. 188–201.

[12] S. Rhea, C. Wells, P. Eaton, D. Geels, B. Zhao, H. Weatherspoon, and J. Kubiatowicz, "Maintenance-free global data storage," *IEEE Internet Comput.*, vol. 5, pp. 40–49, Sept. 2001.

[13] Gnutella [Online]. Available: http://www.gnutella.com

[14] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Security for structured peer-to-peer overlay networks," in *Proc. 5th Symp. Operating System Design and Implementation (USENIX)*, 2002, pp. 299–314.

[15] R. Grimm and J. Nützel, "A friendly peer-to-peer file sharing system with profit but without copy protection," in *Lecture Notes in Computer Science, Innovative Internet Computing Systems*. Heidelberg, Germany: Springer-Verlag, 2002, vol. 2346, pp. 133–142.

[16] Earth Station Five [Online]. Available: http://www.es5.com

[17] M. Kutter, S. Voloshynovskiy, and A. Herrigel, "The watermark copy attack," in *Proc. SPIE—Electronic Imaging 2000, Security and Watermarking of Multimedia Content II*, vol. 3971, 2000, pp. 371–380.

[18] M. van der Veen, T. Kalker, and A. Lemma, "Watermarking and fingerprinting for electronic music delivery," in *Proc. SPIE—Electronic Imaging 2004, Security and Watermarking of Multimedia Content VI*, vol. 5306, 2004, pp. 200–211.

[19] Napster [Online]. Available: http://www.napster.com

[20] Freenet [Online]. Available: http://freenetproject.org

[21] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup service for Internet applications," in *Proc. ACM SIGCOMM*, 2001, pp. 149–160.

[22] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide-area cooperative storage with CFS," in *Proc. 18th ACM Symp. Operating Systems Principles (SOSP)*, 2001, pp. 202–215.

[23] A. Rowstron and P. Druschel, "Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems," in *Lecture Notes on Computer Science, Middleware 2001*, R. Guerraoui, Ed. Heidelberg, Germany: Springer-Verlag, 2001, vol. 2218, pp. 329–350.

[24] B. Y. Zhao, J. Kubiatowicz, and A. D. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing," Comput. Sci. Div., Univ. California, Berkeley, Tech. Rep. CSD-01-1141, 2001.

[25] I. Cox, J. Bloom, and M. Miller, *Digital Watermarking: Principles and Practice*. San Francisco, CA: Morgan Kaufmann, 2002.

[26] SDMI [Online]. Available: http://www.sdmi.org

[27] XrML [Online]. Available: http://www.XrML.org

[28] ODRL [Online]. Available: http://www.odrl.net

[29] R. Pucella and V. Weissman, "A logic for reasoning about digital rights," in *Proc. 15th IEEE Computer Security Foundations Workshop*, 2002, pp. 282–294.

[30] C. N. Chong, R. Corin, S. Etalle, P. H. Hartel, W. Jonker, and Y. W. Law, "LicenseScript: a novel digital rights language and its semantics," in *Proc. 3rd Int. Conf. Web Delivering of Music (WEDELMUSIC)*, K. Ng, C. Busch, and P. Nesi, Eds., 2003, pp. 122–129.

[31] SealedMedia [Online]. Available: http://sealedmedia.com

[32] Microsoft Windows Media [Online]. Available: http://microsoft.com/windows/windowsmedia

[33] Electronic Media Management System [Online]. Available: http://ibm.com/software/emms

[34] openMG [Online]. Available: http://www.openmg.org

[35] Intertrust [Online]. Available: http://www.intertrust.com

[36] ContentGuard [Online]. Available: http://www.contentguard.com

[37] A. M. Eskicioglu and E. J. Delp, "An overview of multimedia content protection in consumer electronics devices," *Signal Process. Image Commun.*, vol. 16, pp. 681–699, 2000.

[38] Light Weight DRM [Online]. Available: http://www.iis.fraunhofer.de/amm/techinf/ipmp/

[39] United States Code [Online]. Available: http://uscode.house.gov

[40] J. Feigenbaum, M. J. Freedman, T. Sander, and A. Shostack, "Privacy engineering for digital rights management systems," in *Lecture Notes in Computer Science, Security and Privacy in Digital Rights Management*, T. Sander, Ed. Heidelberg, Germany: Springer-Verlag, 2001, vol. 2320, pp. 76–105.

Ton Kalker (Fellow, IEEE) received the M.S. and Ph.D. degrees in mathematics from Leiden University, Leiden, The Netherlands, in 1979 and 1986, respectively.

From 1983 to 1985, he was a Lecturer in the Computer Science Department, Technical University of Delft, Since 1985, he has been with the Philips Research Laboratories Eindhoven, Eindhoven, The Netherlands; he is currently a member of the Processing and Architectures for Content MANagement group (PACMAN) of Philips Research, where he is working on security of multimedia content, with an emphasis on watermarking and fingerprinting for video and audio. Since 1999, he has also been a part-time Professor at the Technical University Eindhoven, Eindhoven. Until 1990, he worked in the field of computer-aided design. He specialized in (semi) automatic tools for system verification. He also made contributions to practical applications of watermarking, in particular, watermarking for DVD–video copy protection. His other research interests include wavelets, multirate signal processing, motion estimation, psychophysics, digital video compression, and medical image processing.

Dick H. J. Epema received the M.Sc. and Ph.D. degrees in mathematics from Leiden University, Leiden, The Netherlands, in 1979 and 1983, respectively.

From 1983 to 1984, he was with the Computer Science Department, Leiden University. Since 1984, he has been with the Department of Computer Science, Delft University of Technology, where he is currently an Associate Professor in the parallel and distributed systems group. During the academic year 1987–1988, the fall of 1991, and the summer of 1998, he was also a visiting scientist at the IBM T. J. Watson Research Center, Yorktown Heights, NY. His research interests are in the areas of performance analysis and distributed systems.

Pieter H. Hartel received the M.S. degree in mathematics and computer science from Vrije University, Amsterdam, The Netherlands, in 1978 and the Ph.D. degree in computer science from the University of Amsterdam, Amsterdam, in 1989.

He was with CERN, Geneva, Switzerland; the University of Nijmegen, Amsterdam; and the University of Southampton, Southampton, U.K. He is currently a full Professor of computer science at the University of Twente, Enschede, The Netherlands. His research interests include distributed systems security.

Dr. Hartel is a Member of the Executive Board of CaberNet: Network of Excellence in Distributed and Dependable Computing Systems; Cochair of the program committee of SENTINELS, a future funding program for security research in The Netherlands; Member of the program committee of IOP GenCom, a funding program for generic communication in the private environment; and Founding Member and Chair of IFIP Working Group 8.8 on smart cards.

R. (Inald) L. Lagendijk received the Ph.D. degree from Delft University of Technology, Delft, The Netherlands, in 1990.

Since 1999, he has been Full Professor and Head of the Information and Communication Theory Group at Delft University of Technology. He has coauthored several books on imaging. He is Editor of *Signal Processing: Image Communications*. In recent years he has led or been involved in several research projects, such as "Ubiquitous Communications," Freeband's "Context-Aware Communications, Terminal, and User," STW DIWAMETRIC, and European projects like CERTIMARK and STORit. His research background is in stochastic signal processing and information theory. In particular he is interested in the question how visual information can be represented such that it is not only efficient in communication bandwidth or storage capacity, but that it is also easily accessible when stored in large volumes, that it is robust against errors when transmitted, that it can be used to embed secret information, and that it has a good visual quality. Research projects he is involved in cover subjects such as image and video compression, image quality measures, watermarking, image and video libraries, wireless media streaming, and image sequence restoration and enhancement.

Dr. Lagendijk has been active in various organizational and editorial activities, such as Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING and the organization of the International Conference on Image Processing (ICIP) 2001 (Thessaloniki, Greece) and ICIP 2003 (Barcelona, Spain).

Maarten van Steen received the M.S degree in applied mathematics from the University of Twente, Enschede, The Netherlands, in 1983 and the Ph.D. degree in computer science from Leiden University, Leiden, The Netherlands, in 1988.

He is currently Full Professor in computer science at Vrije University, Amsterdam, The Netherlands, where he conducts research on large-scale distributed systems, notably distributed middleware, content delivery networks, and peer-to-peer systems. He has coauthored an introductory textbook on computer systems and a highly successful textbook on distributed systems that is being used at many colleges and universities worldwide. He has also coauthored more than 70 papers that have appeared in scientific journals and the proceedings of international conferences and workshops.

Dr. van Steen has been Program Committee Member of various international conferences and workshops, Cochair for DOA 2003, and Program Chair for WCW 2004.