

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
22 May 2008 (22.05.2008)

PCT

(10) International Publication Number
WO 2008/060300 A1

(51) International Patent Classification:

G06F 21/00 (2006.01)

(21) International Application Number:

PCT/US2006/060999

(22) International Filing Date:

16 November 2006 (16.11.2006)

(25) Filing Language:

English

(26) Publication Language:

English

(71) Applicant (for all designated States except US): **DYNO-MEDIA, INC.** [US/US]; c/o Cheng Han, 100 Emerson Gardens Rd., Lexington, MA 02420 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **HAN, Cheng** [CN/US]; c/o Cheng Han, 100 Emerson Gardens Rd., Cambridge, MA (US). **WANG, Vincent** [CN/CN]; Beijing (CN). **CHEN, Zhiliang** [CN/CN]; Beijing (CN).

(74) Agent: **LANZA, John, D.**; Choate, Hall & Stewart LLP, Two International Place, Boston, MA 02110 (US).

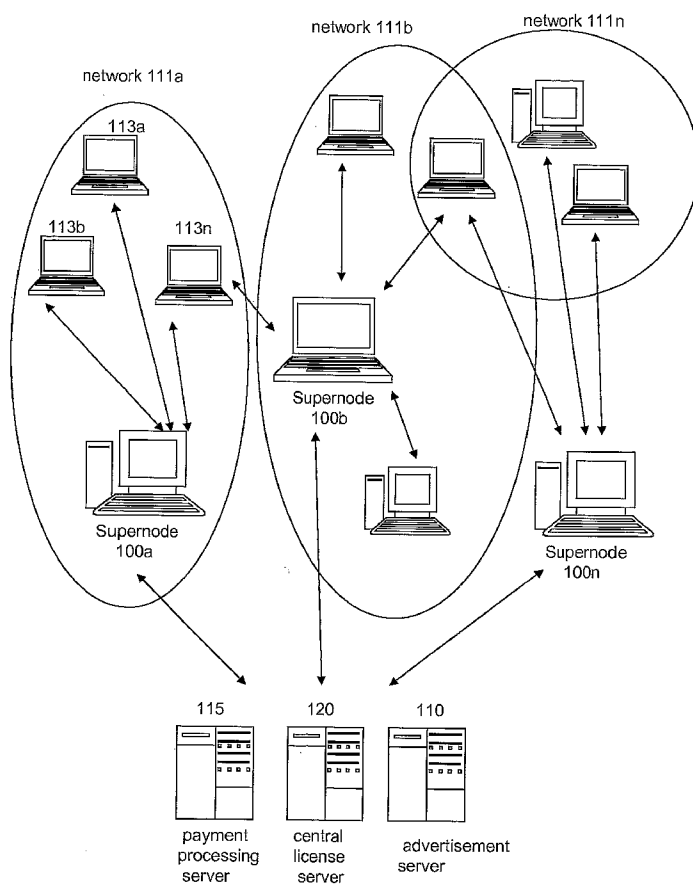
(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report

(54) Title: SYSTEMS AND METHODS FOR DISTRIBUTED DIGITAL RIGHTS MANAGEMENT



(57) Abstract: A digital rights management framework in which license headers containing permissions associated with media files are kept separately from the media files. The license headers may then be stored in a distributed network of peer nodes, which service requests from media players operated by peers to access the media files by consulting the license headers and then, if authorized, transmitting a session key enabling access to the media file. In this manner, the media files may be distributed via any manner, including peer-to-peer transmission, while still enabling flexible, centralized license management. Systems and methods for republishing or media files and distribution of prepaid licenses to media files are also described.

WO 2008/060300 A1

SYSTEMS AND METHODS FOR DISTRIBUTED DIGITAL RIGHTS MANAGEMENT

FIELD OF THE INVENTION

The present invention relates to the field of online media distribution, and specifically to systems and methods for providing digital rights management in a distributed environment.

BACKGROUND OF THE INVENTION

Digital rights management (DRM) schemes are often used by publishers in an attempt to control distribution of digital media files. DRM schemes may be used to control the number of copies of a particular file that may be made, or prevent one or more users from modifying a file. However, existing DRM schemes may face a tradeoff between performance and granular control over distribution. Some DRM schemes may comprise distributing a media file which contains a fixed set of license policies and means for enforcing those policies. While this approach may have some efficiency advantages, it does not allow for the license policies to subsequently be modified without redistributing entirely new copies of the file. Other DRM schemes may require license policies to be obtained from a central server each time a file is accessed. However, this approach can result in a bulky client application, a bad user experience, and performance bottlenecks as many simultaneous requests for a media file may overwhelm the central server. Also, centralized solutions may not be able to offer a variety of licensing options, such as a user prepurchasing a number of licenses for friends.

Peer-to-peer networks have long been available for communicating among computers and sharing files. Peer-to-peer networks have been used among other things, to share and distribute media files among a group of peers. Peer-to-peer networks, as compared to centralized client-server networks, may have the advantage of having fewer bandwidth and processor bottlenecks, as well as being more adaptive to changing network circumstances. However, peer-to-peer networks may be difficult to monitor for purposes of controlling file distribution. Once a file is on a peer-to-peer network, there may no longer be a single access point to the file which a publisher can use to control access. Thus there exists a need for a DRM solution

which combines the efficiency of peer-to-peer networks with the control over licensing policies offered by centralized DRM management.

Further, in the distribution of digital media, advertisements may be used in conjunction with media files in order to generate revenue or product prominence. However, consumers may react negatively if they are compelled to watch advertisements along with a particular media file, and may avoid the file altogether. However, if consumers are given a choice as to whether to watch or listen to an advertisement, so many consumers may choose not to watch the advertisement, that the advertisement does not produce the desired results. Thus there exists a need to target advertising to users of digital media such that user interest in the advertisements displayed or optionally displayed along with a media file generate sufficient results to be successful.

SUMMARY OF THE INVENTION

Broadly, the present invention relates to a digital rights management framework in which license headers containing permissions associated with media files are kept separately from the media files. The license headers may then be stored in a distributed network of peer nodes, which service requests from media players operated by peers to access the media files by consulting the license headers and then, if authorized, transmitting a session key enabling access to the media file. In this manner, the media files may be distributed via any manner, including peer-to-peer transmission, while still enabling flexible, centralized license management.

In one aspect, the present invention is a method for providing collaborative authentication in a distributed digital rights management environment. In one embodiment, the method comprises: receiving, by a server from a first client, a request to access a media file, the request comprising a user identifier, an application identifier, and a media file identifier; determining, by the server, that access to the media file is restricted to a set of users; sending, by the server, a request to a second client identified as having previously interacted with the user identified by the user identifier, the request comprising the user identifier and the application identifier; receiving, by the server, a response from the second client confirming the user identifier corresponds to the application identifier; and transmitting, by the server to the first client, a session key corresponding to the requested media file.

In another aspect, the present invention is a computer-implemented system for providing collaborative authentication in a distributed digital rights management environment. In one embodiment, the system comprises: an authentication server which receives from a first client, a request to access a media file, the request comprising a user identifier, an application identifier, and a media file identifier; determines that access to the media file is restricted to a set of users; sends a request to a second client identified as having previously interacted with the user identified by the user identifier, the request comprising the user identifier and the application identifier; receives a response from the second client confirming the user identifier corresponds to the application identifier; and transmits, to the first client, a session key corresponding to the requested media file.

In a third aspect, the present invention is a computer program product comprising computer readable means for providing collaborative authentication in a distributed digital rights management environment. In one embodiment the product comprises: a media player which executes on a client, and, upon receiving a request from a user to play a media file, sends a request to a server comprising a user identifier, an application identifier, and a media file identifier; and a peer verification module which executes on the client and which receives, from an authentication server, a request comprising a user identifier and an application identifier; determines that the received user identifier corresponds to the application identifier; and transmits a response to the server identifying the determined correspondence.

In a fourth aspect, the present invention is a method of providing targeted advertisements to users viewing media files in a distributed network. In one embodiment the method comprises: receiving, by a server from a user operating a client, a request to access a media file, the request comprising a user identifier identifying the user and a media file identifier identifying the media file; determining, in response to the media file identifier and the user identifier, an advertisement to be displayed to the user in conjunction with the media file; and transmitting, to the client, the determined advertisement and a session key corresponding to the media file.

In a fifth aspect, the present invention is a method of providing targeted advertisements to users viewing media files in a distributed network. In one embodiment the method comprises: receiving, by a server from a user operating a client, a request to access a media file, the request comprising a user identifier identifying the user and a media file identifier identifying the media file; determining,

in response to the media file identifier and the user identifier, an advertisement to be displayed to the user in conjunction with the media file; and transmitting, to the client, the determined advertisement and a session key corresponding to the media file.

In a sixth aspect, the present invention is a computer implemented system for providing targeted advertisements to users viewing media files in a distributed network. In one embodiment the system comprises: an authentication server which receives from a user operating a first client, a request to access a media file, the request comprising a user identifier identifying the user and a media file identifier identifying the media file; transmits the media file identifier and user identifier to an advertisement sever; and transmits to the client a session key corresponding to the media file and an advertisement server, in communication with the authentication server, which determines, in response to a received media file identifier and the user identifier, an advertisement to be displayed to the user in conjunction with the media file; and transmits, to the first client, the determined advertisement.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, aspects, features, and advantages of the invention will become more apparent and may be better understood by referring to the following description taken in conjunction with the accompanying drawings, in which:

FIG. 1 depicts an embodiment of a computer network useful for enabling a distributed digital rights management environment;

FIGs. 2A and 2B depict block diagrams of a typical computer 200 useful as client computing devices and server computing devices;

FIG. 3A depicts a block diagram of an example of a media file access center;

FIG. 3B depicts a flow diagram of a method for providing distributed digital rights management;

FIG. 4A depicts one embodiment of a computer network useful for enabling collaborative authentication in a distributed digital rights management environment;

FIG. 4B depicts a flow diagram of a method for providing collaborative authentication in a distributed digital rights management environment;

FIG. 5 depicts a block diagram of a network for enabling editing and republishing of media files in a distributed digital rights management framework;

FIG. 6 depicts a block diagram of a license header comprising a plurality of sublicenses;

FIG. 7 depicts a flow diagram of one method for providing granular media file permissions in a distributed digital rights management framework;

FIG. 8 depicts a flow diagram of a method for allowing editing and republishing of media files in a distributed digital rights management framework;

FIG. 9 depicts a flow diagram of a method for enabling prepaid purchase of licenses in a distributed digital rights management framework; and

FIG. 10 depicts a flow diagram of a method for providing targeted advertisements to users viewing media files in a distributed network.

DETAILED DESCRIPTION OF THE INVENTION

Now referring to FIG. 1, an embodiment of a computer network useful for enabling a distributed digital rights management environment is shown. In brief overview, a plurality clients 113 in a plurality of networks 111a, 111b, 111n, are in communication with a plurality of supernodes. The supernodes 100, in turn are in communication with one or more central servers 110, 115, 120.

Still referring to FIG. 1, now in greater detail, a computer network useful for enabling distributed digital rights management environment uses a plurality of supernodes to handle requests from a number of clients. The clients may be organized in one or more networks 111a, 111b, 111n, which may comprise any type of network, including without limitation local area networks, wide area networks, and peer-to-peer networks. The requests handled may comprise requests to access a

media file, requests to republish a media file, requests to prepurchase a given number of licenses for a media file, and requests to upload a new media file. The supernodes may be in contact with one or more servers 110, 115, 120, which service any requests which cannot be handled by a supernode.

In some embodiments, the clients may locate a supernode for communication by requesting the network address of a supernode from a centralized server. For example, a central server may maintain an index of available supernodes, and respond to client requests by providing an address of a supernode in proximity to the client making the request. In other embodiments, a client may discover a supernode via communications with peers on the network. In still other embodiments, a client may receive the address of a second supernode via communication with a first supernode. In one embodiment, a client may maintain a table of known supernodes.

In the embodiment shown, one or more of the clients 113 may participate in a peer-to-peer file sharing network. A client 113 may download a media file from a second client 113, and then send a request to a supernode for a session key which will allow the client to play the media file using a media player. The supernodes may be located and selected such that response time to the request is lesser than if all requests for a session key went to a central server.

A server 110, 115, 120 or client 113, 100 may comprise any computing device, including without limitation computing devices such as the ones described in FIGs. 2A and 2B. The client 113 may comprise any device having functionality for playing one or more media files, and sending and receiving information. In some embodiments, the client may comprise software and/or hardware specifically adapted for the playing of media files. In other embodiments, a client may also comprise software and/or hardware comprising a peer verification module which executes on the client. A peer verification module may be used to authenticate requests made by peers with whom a client has communicated with in the past. In one embodiment, a peer verification module may receive, from an authentication server, a request comprising a user identifier and an application identifier; determine that the received user identifier corresponds to the application identifier; and transmit a response to the server identifying the determined correspondence.

In some embodiments, the peer verification module may execute on the client transparently to the user of the client. In one embodiment, the peer verification module may comprise a background process which executes upon a network

connection being established by the client. In another embodiment, the peer verification module may automatically begin executing upon startup of a media file player. In one embodiment, a media file player and a peer verification module may be packaged together for download, or purchase on CD, such that installing the media file player automatically installs the peer verification module as well. In some embodiments, the media file player and the peer verification module may share one or more processes, code, or executables.

A client may also comprise a usage monitor, which operates to monitor the amount and frequency with which the client is online. A usage monitor may also monitor the availability of a client for acting as a file server or as an authentication server.

The clients 113 may be in communication with one or more other clients 113 via peer-to-peer connections. Examples of peer-to-peer interactions may include sharing files, Internet streaming, instant messaging, Voice over Internet Protocol (VoIP) applications, and distributed computing. In one embodiment, a client may store one or more files in a manner such that the files are accessible by one or more other clients. This may be done using any peer-to-peer file sharing or streaming technology. In one embodiment, a number of clients may use a single web site to post links to files and other content the clients are currently sharing.

A supernode 100 may comprise any client or server designated to receive requests from clients 113 to access one or more media files. A supernode may also be referred to as an authentication server. In some embodiments, a supernode may comprise a client 113, with software for handling media file requests. In some embodiments, a supernode may comprise a client which has been selected to serve as a supernode 100 because of certain behavior. Examples of selection criteria for a supernode may comprise reliability thresholds, uptime thresholds, peer verification thresholds, network activity thresholds, connection bandwidth thresholds, and node location algorithms. For example, a client 113 may be elected as a supernode based on participating in a network for a given amount of time. Or, for example, a client 113 may be elected as a supernode based on stability, network speed, or having downloaded or uploaded a certain number of media files.

A supernode may comprise software or hardware which acts as an authentication server, which manages requests from clients 113 to access files and authenticates the clients and one or more users of the clients. In some embodiments,

software comprising functionality for a client to perform supernode functions may be included with a media file player and a peer verification module as described above. In another embodiment, the supernode software may be downloaded by a client upon election of the client as a supernode. In one embodiment, the supernode software may execute transparently to a user of the client. In another embodiment, a user of the client may be prompted to select whether the user wishes the client to perform supernode functions.

A server, such as the servers 110, 115, 120, and the supernode 100 may comprise any computing device or devices capable of sending and receiving information. In some embodiments, a server may comprise a group of servers which act as a logical unit, such as, for example, a server farm or a number of distributed data centers with servers performing related functions. In some embodiments, two or more of servers depicted may reside on the same physical machine. In some embodiments, two or more of the servers depicted may share one or more resources including without limitation processors, memory, and bandwidth.

In some embodiments, a supernode may be in communication with a central license server 120. A central license server may be used as a central repository for licensing information relating to a plurality of media files. In the embodiment shown, the supernode 100 may communicate with a central license server to determine a license that applies to a particular media file. The supernode 100 may also communicate with a central license server to verify the identity of one or more clients.

In some embodiments, the supernode 100 may store information relating to license information to particular media files. In some embodiments, the supernode may store license information relating to previously requested media files, to enable subsequent requests for those media files to be processed more efficiently. In another embodiment, a supernode may receive periodic updates of licensing information relating to media files from a central license server 120. In still other embodiments, a supernode may receive updates from other supernodes 100. The supernodes and central license server or servers may use any techniques for synchronizing license information, including periodic updates, pushed updates, pulled updates, and predictive updates.

In some embodiments, a supernode may also store one or more media files. In other embodiments, a centralized content server may be provided to store media files in the system. In still another embodiment, media files may be stored via a

combination of central servers, supernodes, and clients using peer-to-peer file transfer software.

In the embodiment shown, the supernode 100 is also connected to a payment processing server 115. The payment processing server 115 may comprise any server capable of processing information corresponding to transferring funds between two parties, such as for example, processing credit card charges, credit card credits, bank account withdrawals and bank account deposits. A payment processing server may comprise one or more payment modules including a secured web-service based interface to integrate with micropayment, online payment, mobile payment or legacy payment systems. In some embodiments, a payment processing server may include support for currency conversion, including conversion to one or more virtual currencies used within the system. In some embodiments, the payment processing server 115 may be used to collect revenue associated with one or more purchases of access to a media file. For example, the payment processing server 115 may receive credit card payments from players corresponding to downloading a movie. Or, for example, the payment processing server 115 may distribute funds back to a content publisher. For example, a given audio file might have an associated \$1 download fee. The payment processing server 115 may collect the \$1 fee from a client, and then transfer some or all of the \$1 to an account held by the publisher of the audio file. In some embodiments, a payment processing server may store information relating to one or more user accounts. In these embodiments, users may deposit a given amount of money in an account, and have the account deducted for purchases relating to the system.

In the embodiment shown, the game server 100 is also connected to an advertisement server 110. An advertisement server 110 may comprise any server capable of transmitting one or more advertisements. In some embodiments, an advertisement server may be used to generate targeted advertisements corresponding to a particular media file and end user.

In some embodiments, one or more of the servers discussed may comprise web servers, which may comprise any server capable of delivering content readable by a web browser, including without limitation HTML pages, Javascript, Java applets, Ajax, XML, WML, and images. In some embodiments, the servers may receive and transmit streaming content and services.

The client 113 and the servers may be connected in any manner, and via any network or networks. For example, in some embodiments, the clients 113 may communicate directly with one or more of the supernode 100, the central license server 120, the payment processing server 115, or the advertisement server 110. Connections and networks included in the connections may comprise the Internet, local networks, web servers, file servers, routers, databases, computers, servers, network appliances, or any other computing devices capable of sending and receiving information. The network may comprise computing devices connected via cables, infrared ports, wireless signals, or any other means of connecting multiple computing devices. The network and any devices connected to the networks may communicate via any communication protocol used to communicate among or within computing devices, including without limitation SSL, BitTorrent, HTML, XML, RDP, ICA, FTP, HTTP, SIP, XMPP (also known as Jabber), TCP, IP, UDP, IPX, SPX, NetBIOS, NetBEUI, SMB, SMTP, Ethernet, ARCNET, Fiber Distributed Data Interface (FDDI), RS232, IEEE 802.11, IEEE 802.11a, IEEE 802.11b, IEEE 802.11g and direct asynchronous connections, or any combination thereof. The network may comprise mobile telephone networks utilizing any protocol or protocols used to communicate among mobile devices, including AMPS, TDMA, CDMA, GSM, GPRS or UMTS.

Figures 2A and 2B depict block diagrams of a typical computer 200 useful as client computing devices and server computing devices. As shown in FIGs. 2A and 2B, each computer 200 includes a central processing unit 202, and a main memory unit 204. Each computer 200 may also include other optional elements, such as one or more input/output devices 230a-230b (generally referred to using reference numeral 230), and a cache memory 240 in communication with the central processing unit 202.

The central processing unit 202 is any logic circuitry that responds to and processes instructions fetched from the main memory unit 204. In many embodiments, the central processing unit is provided by a microprocessor unit, such as those manufactured by Intel Corporation of Mountain View, California; those manufactured by Motorola Corporation of Schaumburg, Illinois; the Crusoe and Efficeon lines of processors manufactured by Transmeta Corporation of Santa Clara, California; the lines of processors manufactured by International Business Machines of White Plains, New York; or the lines of processors manufactured by Advanced Micro Devices of Sunnyvale, California.

Main memory unit 204 may be one or more memory chips capable of storing data and allowing any storage location to be directly accessed by the microprocessor 202, such as Static random access memory (SRAM), Burst SRAM or SynchBurst SRAM (BSRAM), Dynamic random access memory (DRAM), Fast Page Mode DRAM (FPM DRAM), Enhanced DRAM (EDRAM), Extended Data Output RAM (EDO RAM), Extended Data Output DRAM (EDO DRAM), Burst Extended Data Output DRAM (BEDO DRAM), Enhanced DRAM (EDRAM), synchronous DRAM (SDRAM), JEDEC SRAM, PC100 SDRAM, Double Data Rate SDRAM (DDR SDRAM), Enhanced SDRAM (ESDRAM), SyncLink DRAM (SLDRAM), Direct Rambus DRAM (DRDRAM), or Ferroelectric RAM (FRAM). In the embodiment shown in FIG. 2A, the processor 202 communicates with main memory 204 via a system bus 250 (described in more detail below). FIG. 2B depicts an embodiment of a computer system 200 in which the processor communicates directly with main memory 204 via a memory port. For example, in FIG. 2B the main memory 204 may be DRDRAM.

FIGs. 2A and 2B depict embodiments in which the main processor 202 communicates directly with cache memory 240 via a secondary bus, sometimes referred to as a “backside” bus. In other embodiments, the main processor 202 communicates with cache memory 240 using the system bus 250. Cache memory 240 typically has a faster response time than main memory 204 and is typically provided by SRAM, BSRAM, or EDRAM.

In the embodiment shown in FIG. 2A, the processor 202 communicates with various I/O devices 230 via a local system bus 250. Various busses may be used to connect the central processing unit 202 to the I/O devices 230, including a VESA VL bus, an ISA bus, an EISA bus, a MicroChannel Architecture (MCA) bus, a PCI bus, a PCI-X bus, a PCI-Express bus, or a NuBus. For embodiments in which the I/O device is an video display, the processor 202 may use an Advanced Graphics Port (AGP) to communicate with the display. FIG. 2B depicts an embodiment of a computer system 200 in which the main processor 202 communicates directly with I/O device 230b via HyperTransport, Rapid I/O, or InfiniBand. FIG. 2B also depicts an embodiment in which local busses and direct communication are mixed: the processor 202 communicates with I/O device 230a using a local interconnect bus while communicating with I/O device 230b directly.

A wide variety of I/O devices 230 may be present in the computer system 200. Input devices include keyboards, mice, trackpads, trackballs, cameras, video cameras, microphones, and drawing tablets. Output devices include video displays, speakers, inkjet printers, laser printers, and dye-sublimation printers. An I/O device may also provide mass storage for the computer system 800 such as a hard disk drive, a floppy disk drive for receiving floppy disks such as 3.5-inch, 5.25-inch disks or ZIP disks, a CD-ROM drive, a CD-R/RW drive, a DVD-ROM drive, tape drives of various formats, and USB storage devices such as the USB Flash Drive line of devices manufactured by Twintech Industry, Inc. of Los Alamitos, California.

In further embodiments, an I/O device 230 may be a bridge between the system bus 250 and an external communication bus, such as a USB bus, an Apple Desktop Bus, an RS-132 serial connection, a SCSI bus, a FireWire bus, a FireWire 800 bus, an Ethernet bus, an AppleTalk bus, a Gigabit Ethernet bus, an Asynchronous Transfer Mode bus, a HIPPI bus, a Super HIPPI bus, a SerialPlus bus, a SCI/LAMP bus, a FibreChannel bus, or a Serial Attached small computer system interface bus.

General-purpose computers of the sort depicted in FIG. 2A and FIG. 2B typically operate under the control of operating systems, which control scheduling of tasks and access to system resources. Typical operating systems include: MICROSOFT WINDOWS, manufactured by Microsoft Corp. of Redmond, Washington; MacOS, manufactured by Apple Computer of Cupertino, California; OS/2, manufactured by International Business Machines of Armonk, New York; and Linux, a freely-available operating system distributed by Caldera Corp. of Salt Lake City, Utah, among others.

For embodiments comprising mobile devices, the device may be a JAVA-enabled cellular telephone, such as the i55sr, i58sr, i85s, or the i88s, all of which are manufactured by Motorola Corp. of Schaumburg, Illinois; the 6035 or the 7135, manufactured by Kyocera of Kyoto, Japan; or the i300 or i330, manufactured by Samsung Electronics Co., Ltd., of Seoul, Korea. In other embodiments comprising mobile devices, a mobile device may be a personal digital assistant (PDA) operating under control of the PalmOS operating system, such as the Tungsten W, the VII, the VIIx, the i705, all of which are manufactured by palmOne, Inc. of Milpitas, California. In further embodiments, the client 113 may be a personal digital assistant (PDA) operating under control of the PocketPC operating system, such as the iPAQ 4155, iPAQ 5555, iPAQ 1945, iPAQ 2215, and iPAQ 4255, all of which

manufactured by Hewlett-Packard Corporation of Palo Alto, California; the ViewSonic V36, manufactured by ViewSonic of Walnut, California; or the Toshiba PocketPC e405, manufactured by Toshiba America, Inc. of New York, New York. In still other embodiments, the mobile device is a combination PDA/telephone device such as the Treo 180, Treo 270, Treo 600, Treo 650, Treo 700, or the Treo 700w, all of which are manufactured by palmOne, Inc. of Milpitas, California. In still further embodiments, the mobile device is a cellular telephone that operates under control of the PocketPC operating system, such as the MPx200, manufactured by Motorola Corp. In still other embodiments, a mobile device may comprise a mobile gaming device with wireless communication capability. A typical mobile device may comprise many of the elements described above in FIG. 2A and 2B, including the processor 202 and the main memory 204.

Referring now to FIG. 3A, a block diagram of an example of a media file access center is shown. In brief overview, a media file access center may comprise a computer application or web page which allows a user to access media files available on a network. A media file access center may comprise means for a user to chat, share media files with, and otherwise communicate with a number of other users or peers. A media file access center 300 may also comprise means for a user to browse, download, and upload media files from one or more centralized locations.

Still referring to FIG. 3A, now in greater detail, in some embodiments, a media file access center 300 may comprise a standalone application. In other embodiments, a media file access center may comprise a web site. A media file access center may be implemented using any programming and/or display languages including without limitation HTML, XML, WML, javascript, Java applets, Ajax, SVG, and Flash.

A media file access center 300 may comprise functionality for a user to browse media files hosted by one or more peers. In some embodiments, a user may be provided with a directory structure in which the user can browse files hosted by peers. In other embodiments, any other interface may be provided, including peer home pages, topic and keyword searches, and searching based on peer recommendations.

A media file access center 300 may also comprise functionality for searching one or more centralized locations for media files. In some embodiments, these central locations may comprise servers which store copies of media files which may also be

hosted on one or more peers. In another embodiment, these central locations may comprise commercial entities which host content.

In some embodiments, a media file access center may be linked or otherwise operate in conjunction with a media file player. For example, a user may locate a media file using the media file access center, and upon selecting the media file, a media file player is launched or activated to play the selected media file. Or for example, a user may select a media file for viewing, and the media file access center may automatically deduct a fee associated with the viewing of the media file from the user's account. The media file access center may then transmit confirmation of the payment and an access key for the media file to a media file player. In other embodiments, a single application may comprise both a media file player and a media file access center.

In some embodiments, the media file access center and/or any content residing on the media file access center may be hosted by a supernode 100. In other embodiments, the media file access center and/or any content residing on the media file access center may be hosted by a centralized server 100.

Referring now to FIG. 3B, one embodiment of a method for providing distributed digital rights management is shown. In brief overview, the method comprises receiving, by a supernode from a client, a request to access a media file protected by a session-based DRM, the request comprising a user identifier, an application identifier, and a media file identifier (step 301); identifying, by the supernode, a license policy corresponding to the media file (step 303); deducting, from an account associated with the user, a payment specified in the license policy with respect to the requested access (step 305); and transmitting, by the supernode to the client, a session key corresponding to the identified file (step 307).

Still referring to FIG. 3B, now in greater detail, one embodiment of a method for providing distributed digital rights management comprises receiving, by a supernode from a client, a request to access a media file protected by a session-based DRM, the request comprising a user identifier, an application identifier, and a media file identifier (step 301). The supernode may receive the request via any protocol or protocols. A media file may comprise any type of media file, including without limitation video, audio, photographic, and multimedia files. Examples of media file protocols include MPEG-3 (also known as MP3), MPEG-4, DivX, XviD, JPEG, GIF,

WINDOWS MEDIA AUDIO defined by Microsoft Corporation, REAL AUDIO as defined by Real Networks.

The media file may be protected by any form of session-based DRM. In one embodiment, the DRM may be specifically coordinated with a media file player program installed on the client. The DRM may be constructed such that only the specifically coordinated media file player program can play the associated media file. In some embodiments, which will be described more fully herein, the session-based DRM may be constructed to operate in conjunction with a unique application identifier, which identifies a specific instance of the media file player program. In these embodiments, a session key which enables a user to play a particular media file would only be valid if used to play the media file using the instance of the media file player program for which the session key was generated.

In some embodiments, the media file may already reside on the client. In other embodiments, the file may reside on the supernode receiving the request, or on a second supernode. In still other embodiments, the file may reside on a central server. In one embodiment, the request to access the media file may also comprise a request to download, stream, or otherwise transfer the file to the client.

A media file identifier may comprise any series of bits and/or bytes that uniquely identifies a media file. A media file identifier may comprise information such as a media file title, author, and length. A media file identifier may comprise a cryptographic hash of some or all of the contents of the media file. In some embodiments, media files uploaded into a media file distribution network may be assigned a media file identifier upon introduction into the network. In some embodiments, this media file identifier may comprise a serial number. In other embodiments, a media file identifier may comprise a real or virtual pathname corresponding to a directory where the media file may be stored. In still other embodiments, a media file identifier may comprise a URL.

In some embodiments, a media file identifier may be unique to every instance of a particular media file. In this embodiment, each time a copy of a media file is made, a new media file identifier is assigned. In this manner individual copies of a media file may be identified and tracked.

A user identifier may comprise any series of bits and/or bytes that uniquely identifies a user. In some embodiments, a user identifier may comprise a screen name, such as the screen names "JeffD" and "SueZQ" shown in FIG 3A. In other

embodiments, a user identifier may comprise a number randomly or pseudo-randomly assigned to a given user. In one embodiment, a user may be assigned a user identifier when the user joins and/or establishes an account with the media file distribution network. In some embodiments, a central server may maintain a database of users and user identifiers.

An application identifier may comprise any series of bits and/or bytes that uniquely identify a media file player. In one embodiment, an application identifier may be created when a media file player is installed on a client. In some embodiments, the application identifier may comprise identifying information corresponding to the client on which the player is installed, such as a CPU identifier, a machine identifier, a hard disk serial number, or a network address. In other embodiments, an application identifier may identify a particular web application or web browser. In some embodiments, a central server may maintain a database of users and user identifiers, and corresponding application identifiers. For example, when a user first joins a media file network, a centralized server may record an identifier for the application the user uses to access media files.

In some embodiments, user and application identifiers may be managed by a central authority, allowing for fine-grained access control to media files. For example, by tracking both application identifiers and user identifiers, a central authority may prevent a user authorized to access a media file from accessing the media file using an application not approved for use with the media file. In this way, unauthorized copying can be minimized by disallowing a media file to be played on any player for which it is not authorized. Or for example, it may be used to prevent two users who may share the same media file player from also sharing one or more media files.

In the method shown, the supernode may then identify a license policy corresponding to the media file (step 303). A license policy may comprise any policy or set of policies which apply to access, use, or distribution of a media file. License policies may for example, define a set of persons who are authorized to access a particular media file. This set of persons may be, for example, a set of friends of the publisher of the media file, a set of people selected for a test screening of the media file, a set of people who have paid a licensing fee for the media file, a set of people who have paid a subscription fee for a series of media files, or a set of people who have had licenses for the media file purchased for them by a third party.

A license policy may also define a set of permissible uses for a media file. For example, a license policy may state that a particular media file is restricted to a single viewing. Or for example, a license policy may indicate that a viewer must pay a given fee for each time the media file is viewed. Or, for example, a license policy may state that a particular media file is restricted to being viewed at a given resolution. In some cases, a license policy may state ways in which a media file may or may not be modified. For example, a license policy may state that a media file may not be altered, but that it may be combined with one or more other media files. Or for example, a license policy may state that the audio of a media file may be altered, but not the video. Or, for example, a license policy may require an author or publishers permission in order to modify the media file. A license policy may also require a fee be paid in conjunction with any particular use or modification of the media file.

A license policy may also define one or more conditions relating to distribution of the media file. For example, a license policy may state that users are free to redistribute a media file. Or for example a license policy may state that users must pay a fee to redistribute a media file, or pay a fee each time someone they redistributed the media file to views the file . Or for example, a license policy may state that users may only redistribute a media file within a certain geographic region. Or for example a license policy may state that users may only redistribute a media file to a certain number of users. Or, for example, a license policy may state that a user may only redistribute a given portion or portions of a media file. Or, for example, a license policy may state that users may only redistribute the media file within larger compilations of media files, or users may only redistribute the media file if it has been modified.

A license policy may comprise any protocol or file format including without limitation HTML, XML, WML, and plain text. In some embodiments, a set of license policies corresponding to a particular media file may be stored in a single file. In other embodiments, a single file may comprise a set of license policies which apply to more than one media file. In some embodiments, some or all of a license policy may be stored using encryption and/or digital signatures such that the license policy can be authenticated and cannot be tampered or altered. In some embodiments, a license policy may be included in a media file header, which comprises other information relating to a media file.

In some embodiments, the supernode may identify a license policy corresponding to the media file (step 303) by searching a database of license policies stored on the supernode. This database may comprise any means of storing license policies. In cases where the license policy does not exist on the supernode, the supernode may send a request to a central license server for a license policy. In some embodiments, the supernode may receive periodic updates of license policies from one or more central license servers. In other embodiments, the supernode may contact one or more publishers of the media file to identify a license policy corresponding to the media file.

In the embodiment shown, the method then comprises deducting, from an account associated with the user, a payment specified in the license policy with respect to the requested access (step 305). In some embodiments, the deduction may be accomplished by means of a payment processing server as discussed herein.

In the embodiment shown, the method then comprises transmitting, by the supernode to the client, a session key corresponding to the identified file (step 307). This transmission may be done using any protocol or protocols. In some embodiments, one or more media files may be transmitted in addition to the session key. In another embodiment, one or more advertisements may be transmitted in addition to the session key.

Referring now to FIG. 4A, one embodiment of a computer network useful for enabling collaborative authentication in a distributed digital rights management environment is shown. In brief overview, a number of clients 113a, 113b, 113n (generally 113) are connected to a designated client 100, referred to as a supernode. The supernode receives, from a first client, a request to access a media file, which may comprise a user identifier, an application identifier, and a media file identifier. The supernode may determine, by consulting a file directory or a central license server 120 that access to the media file is restricted to a set of users. The supernode may then send a request to a number of clients 113b, 113n to verify that the application identifier provided by the client 113a corresponds to the user identifier provided. The supernode may then receive a response from the clients 113b, 113n confirming the user identifier corresponds to the client address. The supernode may then transmit a session key allowing the client 113a to access the identified media file. The supernode may also transmit an advertisement from an advertisement server 120

along with the session key, and may communicate with a payment processing server 110 if payment is required to access the media file.

Referring now to FIG. 4B, one embodiment of a method for providing collaborative authentication in a distributed digital rights management environment is shown. In brief overview, the method comprises: receiving, by a server from a first client, a request to access a media file, the request comprising a user identifier, an application identifier, and a media file identifier (step 401); determining, by the server, that access to the media file is restricted to a set of users (step 403); sending, by the server, a request to a second client identified as having previously interacted with the user identified by the user identifier, the request comprising the user identifier and the client address (step 405); receiving, by the server, a response from the second client confirming the user identifier corresponds to the client address (step 407); and transmitting, by the server to the first client, a session key corresponding to the requested media file (step 409) and sending, to a second server, a request to deduct a given amount from an account associated with the user identifier (step 411). The method may be used by a supernode to validate a client request without the need to send a request to a central server to confirm that the received application identifier and user identifier match.

Still referring to FIG. 4B, now in greater detail, the method shown comprises receiving, by a server from a first client, a request to access a media file, the request comprising a user identifier, an application identifier, and a media file identifier (step 401). The server may comprise any server described herein, including without limitation a supernode 100. The user identifier, application identifier, and media file identifier may comprise any identifiers described herein. In some embodiments, the request may be transmitted from a media file access center.

In the embodiment shown, the method then comprises determining, by the server, that access to the media file is restricted to a set of users (step 403). In one embodiment, the server may determine that access to the media file is restricted by identifying a license policy associated with the media file. This identification may be done by any means described herein, including without limitation searching a local directory, and making a request to a central license server.

In the embodiment shown, the method then comprises sending, by the server, a request to a second client identified as having previously interacted with the user identified by the user identifier, the request comprising the user identifier and the

application identifier (step 405). This step may be performed in cases where the server has no record of the particular application identifier corresponding to the user identifier. This may occur in cases where a user has recently installed a new media file player, or is using a computer of a friend. Alternatively, this scenario may occur if a malicious user is attempting to gain access to a media file under the guise of another user. The server may send the request using any protocol or protocols. In some embodiments, the server may send a request to a plurality of clients identified as having previously interacted with the user identified by the user identifier. In some embodiments, the request may be sent to a peer verification module executing on the second client as described herein.

A server may use any method to identify a second client as having previously interacted with the user identified by the user identifier. In some embodiments, the server may keep a record of past client interactions. For example, a server might receive a request from a user to access a particular media file, and the request may indicate that the media file is hosted by a given client. The server could then infer that the user and client have been in communication. In other embodiments, the server may send a request to a number of clients inquiring whether they have previously interacted with the user. For example, the server might send a request to all clients on the same LAN as the user. Or, for example, the server might send a request to all clients identified by the user as "friends" or "contacts" in a buddy list. In still other embodiments, clients may periodically update the supernode of users with whom the clients have communicated.

After receiving the request, the client may determine, by checking any records stored on the client, whether the client has communicated with the identified user, and if so, whether the user was communicating using the identified application. Examples of records of communication with the user and the application may include chat logs, file transfer records, and blog or message board postings. The client may then respond to the server indicating whether the client has any record of the identified user using the identified application.

In the embodiment shown the server may then receive a response from the second client confirming the user identifier corresponds to the client address (step 407). In some embodiments, the server may receive a plurality of responses from a plurality of clients.

In some embodiments, the server may use a predetermined threshold to determine whether to accept the peer verification. In one embodiment, the server may require that at least a given number of positive confirmations be received in order to accept the verification. For example, the server may require that positive verifications be received from at least two clients. In other embodiments, the server may require that a certain percentage of the responses received by the server be positive confirmations. For example, the server may require at least 75% of the received responses to include positive verifications in order to accept the request. If the threshold is not met, the server may either send out more requests to other clients, send a request to a central server to authenticate the user identifier/application identifier pair, or simply deny the request.

After the server determines the user is authorized to view the media file, and authenticates the user and application identifiers, the server may then transmit to the first client, a session key corresponding to the requested media file (step 409). In some embodiments, the server may also transmit one or more advertisements, as will be discussed herein with respect to FIG. 10.

In the embodiment shown, the method also comprises sending, to a second server, a request to deduct a given amount from an account associated with the user identifier (step 411). This request may be sent using any protocol or protocols. In some embodiments, the second server may comprise a payment processing server. In other embodiments, the second server may comprise a supernode. In some embodiments, the server may wait until the server receives confirmation of the deduction before transmitting the session key to the client.

Referring to FIG. 5, a block diagram of a network for enabling editing and republishing of media files in a distributed digital rights management framework is shown. In brief overview, a client 113a (referred to as the republisher) transmits a request to modify and/or redistribute a given media file registered to a publisher associated with a second client 113e (referred to as the publisher). The request is passed through a supernode and potentially a number of servers, and arrives at the publisher. The publisher may then respond to the request, either with an automated response or by presenting the request to a user. The response is passed through one or more servers, and the permission arrives at the client. After receiving the permission the user may modify and/or redistribute the media file.

Now referring to FIG. 6, a block diagram illustrating a license header comprising a plurality of sublicenses is shown. In brief overview, a license header 600 comprises a plurality of sublicenses 620a, 620b, 620c (generally 620). Each sublicense corresponds to one or more chapters 625a, 625b, 625c, 625d (generally 625) of the media file.

Still referring to FIG. 6, now in greater detail, a license header may comprise a number of sublicenses. A license header may comprise any protocol or protocols, including XML, HTML, and WML. A license header may correspond to one or more media files.

In some embodiments, a license header may be digitally signed by an entity so as to enable a client or a media file player to verify the authenticity of the license header. In some embodiments, a license header may be signed by a central authority which is known and trusted by the media file player. In other embodiments, a license header may be digitally signed by the creator of the media file.

A license header may comprise one or more license policies, as described herein. In some embodiments, different chapters of a media file may be associated with different sublicenses and, consequently, different license policies. A chapter of a media file may comprise any discrete unit of a media file. A chapter may comprise a temporal unit, such as the first 30 seconds of a video clip. A chapter may comprise a spatial unit, such as a logo displayed in the bottom right corner of a video clip. A chapter may comprise a media unit, such as the soundtrack of a video clip. A chapter may comprise a subgroup of a particular media element, such as the bass line in a music recording.

A license header may be specific to a single instance of a media file. For example, a publisher may release 10 copies of the same music video. Each copy may have a unique media file identifier and a unique license header, each license header referring only to the media file identifier associated with it. In this manner, different copies of a media file may have different license policies. This allows a publisher to exert control over individual copies. For example, a publisher may distribute a copy to a friend which allows for unlimited use and modification, but distribute a copy to a newspaper reviewer which allows for only a limited number of viewings and no modifications.

A license header may also correspond to a plurality of copies of a media file. For example, a license header declaring that a media file may freely be distributed

may correspond to each copy of the media file that is made. In some embodiments this may be accomplished by duplicating the media file identifier along with the media file when a copy is made, rather than creating a new media file identifier. In other embodiments, this may be accomplished by creating a license header which applies to media files having media file identifiers within a certain range or with a certain prefix. A license header might then specify that the license applies to all videos having the prefix "BobsVideo0001". Each copy of BobsVideo0001 may then be created with the media file identifier having the form of "BobsVideo0001*****". In this way, individual copies may be tracked separately, while requiring only a single license header to be maintained. In other embodiments, a media file identifier and a copy identifier, identifying a particular copy of the media file, may be included as separate parameters.

A license header may be stored either together or separately from its corresponding media file. In some embodiments, supernodes may store and manage requests for license headers, while clients store the media files themselves. The media files could be protected with DRM such that they cannot be played except in a media file player which requires a license header authorizing the playing. In this manner, although a user could freely download files from other users without regard to licensing, in order to play or make use of the media file, the user would have to obtain a license header which authorized the user and the media file player to access the media file.

In the license header 600 shown, the license comprises three sublicenses applying to four chapters. The first 620a and third 620c sublicenses correspond to content imported by the creator of the media file from other publishers. The second 620b sublicense corresponds to content created by the creator of the combined media file. The "inherited tag" indicates that the license that applies to the chapters 2 & 3 is also the license that applies to the combined work as a whole.

In some embodiments, a license header may incorporate multiple DRM schemes. For example, the license header as a whole may be digitally signed by one authority, and the combined media file may be protected with DRM applicable to one media file player. One of the sublicenses might designate a given chapter as protected by an alternative and/or proprietary DRM scheme.

A media file player may be adapted specifically to recognize license headers and apply any license policies contained in the header. In cases where a license

header denotes that a media file comprises more than one DRM scheme, the media file player may interface with one or more additional media file players in order to play the content protected by any DRM schemes that are not recognized or implemented by the first media player. In some embodiments, a license header may comprise information relating to a particular media file that can be used to play a given chapter of the corresponding media file.

For example, a publisher may create a video that the publisher intends for free public distribution, but using a soundtrack that is protected by a proprietary DRM format. The license header may comprise information relating to the DRM format of the soundtrack. After a user downloads the video from a site of the publisher, the user's media file player may then read the license header, and discover whether the user has the required second media file player for playing the soundtrack. The media file player may then either prompt the user to install the second player, or launch the second player automatically. If the soundtrack requires the user purchase a license, the either the first or second media file player may prompt the user as to whether the user would like to purchase the license, and provide means for doing so. The two players may then work in conjunction to play the video along with the soundtrack. Or for example, a video file may contain one video track (V1), one free audio track (A1) with low quality, or limited number of playback, and one pay-for audio track (A2). The media file player may allow consumers to play the video file by 1 media file player, and if the user does not pay, the user can access the free or free-preview sound track. Therefore, the media file publisher can choose to distribute the video file (V1+A1) to the public. After the preview, users may choose to purchase and download the high-quality sound track (A2).

In some embodiments, the license header may comprise information which allows multiple media file players to coordinate playback. For example, if a song used as part of a video soundtrack requires a separate media file player, the license header may identify both the media file player needed and the time and duration at which the song occurs. In this manner the primary media file player may launch or otherwise activate the needed media file player at the correct time.

Now referring to FIG. 7, a flow diagram illustrating one method for providing granular media file permissions in a distributed digital rights management framework is shown. In brief overview, the method comprises creating a combined media file, the combined media file comprising a plurality of media elements, wherein each

element comprises at least one DRM protection (step 701); creating a license header corresponding to the combined media file, wherein the license header comprises licensing information corresponding to both the combined media file and each of the plurality of media elements (step 703); and associating the license header with the combined media file (step 705).

Still referring to FIG. 7, now in greater detail, the method comprises creating a combined media file, the combined media file comprising a plurality of media elements, wherein each element comprises at least one DRM protection (step 701). Each media element may comprise any type of media file. In some cases the media elements may each be protected with the same DRM scheme, in other cases each media element may be protected with a different DRM scheme. The combined media file may be created by a media file editor designed to interact with DRM-protected media files, and specifically media files with license headers as described herein. In one embodiment, the media file editor may be a standalone application. In other embodiments, the media file editor may be integrated with a media file player and/or a media file access center as described herein. In some embodiments, the media file editor may comprise a web-based application.

The method shown then comprises creating a license header corresponding to the combined media file, wherein the license header comprises licensing information corresponding to both the combined media file and each of the plurality of media elements (step 703). This license header may comprise any license header described herein, and the licensing information may comprise any licensing policy or policies described herein. The license header may also contain any other information relating to the media file, including author and publisher information, a media file identifier.

The method shown then comprises associating the license header with the combined media file (step 705). This association may be done by incorporating a media file identifier corresponding to the media file into the license. In some embodiments, the association may comprise digitally signing the both the license header and the corresponding media file. The combined media file may then be distributed to other clients and users using any of the means described herein.

Now referring to FIG. 8, a flow diagram illustrating a method for allowing editing and republishing of media files in a distributed digital rights management framework is shown. In brief overview, the method comprises: transmitting, by a client to a server, a request to republish a first media file, the media file having an

associated license (step 801); receiving, from the server, a permission to republish the first media file (step 803); and creating a second media file, the second media file comprising a portion of the first media file, and comprising a license header with license information corresponding to the first media file and license information corresponding to the second media file (step 805).

Still referring to FIG. 8, now in greater detail, the method comprises transmitting, by a client to a server, a request to republish a first media file, the media file having an associated license (step 801). The request may be transmitted by any protocol or protocols. In some instances, the request may be transmitted to a supernode 100. In some embodiments, the request may be transmitted by a media file player. In other embodiments, the request may be transmitted by a media file editor.

A request to republish a music file may comprise a request to redistribute, or a request to modify and/or incorporate into a compilation and redistribute. In some cases, the request may specify one or more license policies which will apply to the redistribution, such as that the redistribution will be limited to a certain number of users, or that the users will have to pay a fee.

In cases where the request is received by a supernode, the supernode may consult a local database of license policies to determine if the license policy associated with the media file permits the request. If no license policy is found locally, the supernode may send a request to a central license server to determine whether the license policy associated with the media file permits the request. In the event the request is not permitted, or in the event no license is found, the supernode or the central license server may forward the request to the media file publisher 113e. In some cases the request may be forwarded through one or more other supernodes. The publisher 113e may then either respond automatically to the forwarded request, or present the request to a user, who may respond.

The client may then receive, from the server, a permission to republish the first media file (step 803). In some embodiments, the permission may comprise a license header with a license policy indicating the permission. For example, a client may transmit a request to redistribute a video clip in conjunction with an advertisement or in conjunction with a hint to play an advertisement, the video clip having a given media file identifier. The client may then receive a permission comprising a license header identifying the media file identifier, and comprising a license policy allowing the user to redistribute the video clip under certain conditions.

Since the media file identifier is unique to the specific instance of the video clip, the license header will not be able to be transferred to other copies of the video clip currently held by other users. Further, the license header and media file may be suitably encrypted and digitally signed by the publisher or a trusted third party so as to prevent tampering by users or redistributors. In this way a publisher can achieve fine-grained control over redistribution of media files.

The client may then create a second media file, the second media file comprising a portion of the first media file; and comprising a license header with license information corresponding to the first media file and license information corresponding to the second media file (step 805). The second media file may be created by a media file editor specifically designed for creating such files. The media file editor may read the permission so as to prevent the client from violating any license policies concerning the redistribution. For example, in response to a license policy, the media file editor may limit the total amount of material the user can incorporate in the created media file to be no longer than 15 seconds. Or, for example, the media file editor may prevent the user from modifying the created media file. Or, for example, the media file editor may prevent the user from removing certain portions of the first media file, such as a copyright notice, or closing credits.

The above method may also be applied to any other request from a client relating to a media file, including requests to modify the media file and requests to modify the media file and then redistribute the media file.

Referring now to FIG. 9, a flow diagram of a method for enabling prepaid purchase of licenses in a distributed digital rights management framework is shown. In brief overview, the method comprises: receiving, at a server from a client, a request to purchase a quantity of prepaid licenses for a media file, the request comprising a user identifier (step 901); creating, at the server, a copy of the requested media file, wherein the copy is encoded with a license indicating the quantity of prepaid licenses (step 903); associating, by the server, a file identifier with the copy (step 905); and transmitting, by the server to the client, the file identifier (step 907); receiving, by a second server from a second client, a request to access a media file, the request comprising the file identifier (step 909); determining, by the second server, that the quantity of prepaid licenses associated with the identified file is at least one (step 911); decrementing, by the second server, the quantity of prepaid licenses associated

with the identified file (step 913); and transmitting, by the second server to the second client, a session key corresponding to the identified file (step 915).

Still referring to FIG. 9, now in greater detail, the method comprises: receiving, at a server from a client, a request to purchase a quantity of prepaid licenses for a media file, the request comprising a user identifier (step 901). In some embodiments the server may comprise a supernode, in other embodiments, the server may comprise a central license server. The request for prepaid licenses may comprise a request for any type of license, including a license to access, modify, or redistribute. In some embodiments, the request may specify one or more user identifiers corresponding to users who the prepaid licenses are intended for.

The method may then comprise creating, at the server, a copy of the requested media file, wherein the copy is encoded with a license indicating the quantity of prepaid licenses (step 903). In some embodiments the server may communicate with a payment processing server to deduct the amount of the prepaid licenses from an account corresponding to the user identifier.

The copy of the media file may be stored in any location. In some embodiments, the copy may be stored on the server. In other embodiments, the copy may be stored on one or more clients. In still other embodiments, the copy may be stored on a plurality of supernodes.

The server may also create a license header for the created copy of the media file. The license header may specify the quantity and nature of the prepaid licenses.

The method may then comprise associating, by the server, a file identifier with the copy (step 905).. The file identifier may comprise any identifier which identifies the location of the file. In some embodiments, the file identifier may comprise a URL. In other embodiments, the file identifier may comprise a media file identifier as described herein. The method may then comprise transmitting, by the server to the client, the file identifier (step 907). The client may forward the file identifier to one or more other clients who the client wishes to distribute the prepaid licenses to. Those other clients may in turn transmit a request for the file. This request may be transmitted by a media file player, a media file editor, or a web browser in cases where the file identifier is a URL.

In the embodiment shown, the method then comprises receiving, by a second server from a second client, a request to access a media file, the request comprising the file identifier (step 909). In some embodiments, the second server may comprise

the same server as the first server. In other embodiments, they may comprise two different servers.

The second server may then determine that the quantity of prepaid licenses associated with the identified file is at least one (step 911). In some embodiments, the second server may make this determination by identifying a license header corresponding to the media file pointed to by the file identifier. The second server may identify this license header via any method described herein, including consulting a local database, or sending a request to another server. If the quantity is less than one, this may indicate that all prepaid licenses have been used, and thus the request should be denied. In some cases the server may also check to see if the user making the request for the media file is a user authorized by the prepaid licenses.

If the second server determines the request is valid the second server may then decrement the quantity of prepaid licenses associated with the identified file (step 913). The second server may then transmit, to the second client, a session key corresponding to the identified file (step 915). The second sever may also transmit a copy of the media file or transmit a location from which the media file may be downloaded or streamed.

Referring now to FIG. 10, a flow diagram of a method for providing targeted advertisements to users viewing media files in a distributed network is shown. In brief overview, the method comprises: receiving, by a server from a user operating a client, a request to access a media file, the request comprising a user identifier identifying the user and a media file identifier identifying the media file (step 1001); determining, in response to the media file identifier and the user identifier, an advertisement to be displayed to the user in conjunction with the media file (step 1003); and transmitting, to the first client, the determined advertisement and a session key corresponding to the media file (step 1005).

Still referring to FIG. 10, now in greater detail, method for providing targeted advertisements comprises receiving, by a server from a user operating a client, a request to access a media file, the request comprising a user identifier identifying the user and a media file identifier identifying the media file (step 1001). This step may be performed in accordance with any of the embodiments described herein. In some embodiments, the request may also comprise an application identifier.

The method may then comprise determining, in response to the media file identifier and the user identifier, an advertisement to be displayed to the user in

conjunction with the media file (step 1003). In some embodiments, the server may determine the advertisement to display without contacting an additional server. In other embodiments, the server may send a request to a second server, such as an advertisement server, which may determine the advertisement to be displayed. In some embodiments, the server may determine an advertisement to be displayed also in response to an application identifier received.

An advertisement may comprise any form of advertisement. In some embodiments, an advertisement may comprise another media file of either the same or a different format than the requested media file. In some embodiments, an advertisement may comprise a media file and an associated license header.

In one embodiment, the server determines based on at least one media file previously accessed by the user, an advertisement to be displayed to the user in conjunction with the media file. The server may identify an advertisement with a common theme or a shared keyword with the previously viewed media file. The server may identify an advertisement published by a publisher who also published a media file previously accessed by the user.

In another embodiment, the server determines, based on at least one advertisement previously accessed by the user, an advertisement to be displayed to the user in conjunction with the media file. The server may identify an advertisement with a common theme or a shared keyword with the previously viewed advertisement. The server may identify an advertisement published by a publisher who also published an advertisement previously accessed by the user.

In still another embodiment, the server determines, based on a length of time a previous advertisement was played by the user, an advertisement to be displayed to the user in conjunction with the media file. In this manner, the server may identify a previous advertisement which held the user's interest. In this manner, a server may also identify advertisements which did not hold the user's interest—those that the user closed or stopped after a brief period. Again, the server may use any information corresponding to the previously viewed advertisement in determining the advertisement to be shown.

In still another embodiment, the server determines, based on a response to a previous advertisement by the user, an advertisement to be displayed to the user in conjunction with the media file. A user response may comprise any user action with respect to the advertisement, including a user clicking on a link within the

advertisement, a user purchasing an item via the advertisement, a user forwarding the advertisement to one or more other users, a user playing the advertisement multiple times, or a user muting an advertisement. Again, the server may use any information corresponding to the previously viewed advertisement in determining the advertisement to be shown.

In one embodiment, the server determines, based on at least one advertisement previously displayed in conjunction with the context of the media file, an advertisement to be displayed to the user in conjunction with the media file. The server may identify advertisements that are particularly successful or unsuccessful when shown alongside certain videos, and then attempt to show advertisements that have been successful at generating positive user responses along with future requests for the media file. The server may measure whether an advertisement was successful with another viewer of the media file using any metric, including without limitation time of viewing, response to a link, purchase of an item, replaying the advertisement, or forwarding the advertisement to another user.

The method may then comprise transmitting, to the client, the determined advertisement and a session key corresponding to the media file. In some embodiments the determined advertisement may be transmitted by the server which received the request. In other embodiments, the determined advertisement may be transmitted by an advertisement server. In still other embodiments, the determined advertisement may be transmitted by a second client. In some embodiments, a server may transmit a link from which the client can download the determined advertisement.

In some embodiments, the user may be forced to watch the determined advertisement prior to playing the requested media file. In other embodiments, the user may be able to view the advertisement at the user's option.

While the invention has been particularly shown and described with reference to specific preferred embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein departing from the spirit and scope of the invention as defined by the appended claims.

We Claim:

1. A method for providing collaborative authentication in a distributed digital rights management environment, the method comprising:
 - (a) receiving, by a server from a first client, a request to access a media file, the request comprising a user identifier, an application identifier, and a media file identifier;
 - (b) determining, by the server, that access to the media file is restricted to a set of users;
 - (c) sending, by the server, a request to a second client identified as having previously interacted with the user identified by the user identifier, the request comprising the user identifier and the application identifier;
 - (d) receiving, by the server, a response from the second client confirming the user identifier corresponds to the application identifier; and
 - (e) transmitting, by the server to the first client, a session key corresponding to the requested media file.
2. The method of claim 1, wherein the application identifier comprises an address corresponding to the client.
3. The method of claim 1, wherein the application identifier comprises a value determined upon installation on the client of a media player application.
4. The method of claim 1, wherein step (b) comprises determining, by the server, that the media file is restricted to a set of users, the set of users comprising one of the following:
 - a set users who have paid an access fee,
 - a set of users who reside in a given geographic area, and
 - a set of users identified as trusted users by the publisher of the media file.
5. The method of claim 1, wherein step (b) comprises determining, by the server, based on a locally stored access policy for the media file, that the media file is restricted to a set of users.

6. The method of claim 1, wherein step (b) comprises the steps of:
 - (b-a) sending a request to a second server, the request comprising the media file identifier; and
 - (b-b) receiving a response from the second server, the response comprising an access policy for the identified media file.
7. The method of claim 1, wherein step (c) comprises sending, by the server, a request to a second client identified as having previously communicated with the user identified by the user identifier, the request comprising the user identifier and the application identifier.
8. The method of claim 1, wherein step (c) comprises sending, by the server, a request to a second client identified as having previously corresponded with the user identified by the user identifier, the request comprising the user identifier and the application identifier.
9. The method of claim 1, wherein step (c) comprises sending, by the server, a plurality of requests to a plurality of clients identified as having previously interacted with the user identified by the user identifier, the requests comprising the user identifier and the application identifier.
10. The method of claim 9, wherein step (d) comprises receiving, by the server, a plurality of responses from plurality of clients, wherein at least one of the responses confirms the user identifier corresponds to the application identifier.
11. The method of claim 9, wherein step (d) comprises the steps of:
 - (d-a) receiving, by the server, a plurality of responses from plurality of clients, wherein at least one of the responses confirms the user identifier corresponds to the application identifier.
 - (d-b) determining that the number of responses confirming that the user identifier corresponds to the application identifier exceeds a given threshold

12. The method of claim 1, further comprising the step of: sending, to a second server, a request to deduct a given amount from an account associated with the user identifier.
13. A computer-implemented system for providing collaborative authentication in a distributed digital rights management environment, the system comprising:
 - an authentication server which receives from a first client, a request to access a media file, the request comprising a user identifier, an application identifier, and a media file identifier; determines that access to the media file is restricted to a set of users; sends a request to a second client identified as having previously interacted with the user identified by the user identifier, the request comprising the user identifier and the application identifier; receives a response from the second client confirming the user identifier corresponds to the application identifier; and transmits, to the first client, a session key corresponding to the requested media file.
14. The system of claim 13, wherein the application identifier comprises an address corresponding to the client.
15. The system of claim 13, wherein the application identifier comprises a value determined upon installation on the client of a media player application.
16. The system of claim 13, wherein the authentication server determines that the media file is restricted to a set of users, the set of users comprising one of the following:
 - a set users who have paid an access fee,
 - a set of users who reside in a given geographic area, and
 - a set of users identified as trusted users by the publisher of the media file.
17. The system of claim 13, wherein the authentication server determines, based on a locally stored access policy for the media file, that the media file is restricted to a set of users.

18. The system of claim 13, wherein the authentication server sends a request to a second server, the request comprising the media file identifier; and receives a response from the second server, the response comprising an access policy for the identified media file.
19. The system of claim 13, wherein the authentication server sends a request to a second client identified as having previously communicated with the user identified by the user identifier, the request comprising the user identifier and the application identifier.
20. The system of claim 13, wherein the authentication server sends a request to a second client identified as having previously corresponded with the user identified by the user identifier, the request comprising the user identifier and the application identifier.
21. The system of claim 13, wherein the authentication server sends a plurality of requests to a plurality of clients identified as having previously interacted with the user identified by the user identifier, the requests comprising the user identifier and the application identifier.
22. The system of claim 21, wherein the authentication server receives a plurality of responses from plurality of clients, wherein at least one of the responses confirms the user identifier corresponds to the application identifier.
23. The system of claim 21, wherein the authentication server: receives a plurality of responses from plurality of clients, wherein at least one of the responses confirms the user identifier corresponds to the application identifier; and determines that the number of responses confirming that the user identifier corresponds to the application identifier exceeds a given threshold
24. The system of claim 13, wherein the authentication server sends, to a second server, a request to deduct a given amount from an account associated with the user identifier.

25. A computer program product comprising computer readable means for providing collaborative authentication in a distributed digital rights management environment, the product comprising:
- a media player which executes on a client, and, upon receiving a request from a user to play a media file, sends a request to a server comprising a user identifier, an application identifier, and a media file identifier; and
 - a peer verification module which executes on the client and which receives, from an authentication server, a request comprising a user identifier and an application identifier; determines that the received user identifier corresponds to the application identifier; and transmits a response to the server identifying the determined correspondence.
26. The computer program product of claim 25, wherein the peer verification module executes transparently to the user.
27. The computer program product of claim 25, wherein the media player receives, from a server, a session key corresponding to the media file; decodes the media file using the session key, and plays the media file.
28. The computer program product of claim 25, wherein the media player receives, from a server, a session key corresponding to a portion of a streaming media file, decodes the portion of a media file using the session key, and plays the portion of the media file.
29. The computer program product of claim 25, wherein the media player receives, from a server, a session key corresponding to a portion of a streaming media file, decodes the portion of a media file using the session key, and plays the portion of the media file.
30. The computer program product of claim 25, wherein the media player receives, from a server, a session key corresponding to a portion of a streaming media file, decodes the portion of a media file using the session key;

plays the portion of the media file; identifies a second session key in the decoded media file; receives a second portion of the media file, and decodes the portion of a media file using the second session key.

31. The computer program product of claim 25, wherein the peer verification module determines that the received user identifier corresponds to the application identifier based on at least one previous instance of receiving the user identifier in conjunction with a connection with a client using the application identifier.
32. The computer program product of claim 25, wherein the peer verification module determines that the received user identifier corresponds to the application identifier based on at least one previous instance of receiving a communication from a client using the application identifier and including the user identifier and comprising a media file.
33. The computer program product of claim 25, wherein the peer verification module determines that the received user identifier corresponds to the application identifier based on at least one previous instance of receiving a communication from a client using the application identifier and including the user identifier and comprising an instant message.
34. The computer program product of claim 25, wherein one or more executable files comprising the peer verification module are integrated with one or more executable files comprising the media player.
35. The computer program product of claim 25, wherein the peer verification module maintains a persistent connection with one or more authentication servers.
36. The computer program product of claim 25, further comprising a usage monitor.

37. The computer program product of claim 25, further comprising a usage monitor which stores information relating to an amount and frequency of use of the media player.
38. The computer program product of claim 25, further comprising a usage monitor which stores information relating to a percentage of requests from an authentication server which are received and responded to.
39. The computer program product of claim 25, further comprising: an authentication server which receives from a first client, a request to access a media file, the request comprising a user identifier, an application identifier, and a media file identifier; determines that access to the media file is restricted to a set of users; sends a request to a second client identified as having previously interacted with the user identified by the user identifier, the request comprising the user identifier and the client address; receives a response from the second client confirming the user identifier corresponds to the client address; and transmits, to the first client, a session key corresponding to the requested media file.
40. The computer program product of claim 39, wherein the authentication server executes transparently to the user.
41. The computer program product of claim 39, wherein the authentication server determines that the media file is restricted to a set of users, the set of users comprising one of the following:
- a set users who have paid an access fee,
 - a set of users who reside in a given geographic area, and
 - a set of users identified as trusted users by the publisher of the media file.
42. The computer program product of claim 39, wherein the authentication server determines, based on a locally stored access policy for the media file, that the media file is restricted to a set of users.

43. The computer program product of claim 39, wherein the authentication server sends a request to a second server, the request comprising the media file identifier; and receives a response from the second server, the response comprising an access policy for the identified media file.
44. The computer program product of claim 39, wherein the authentication server sends a request to a second client identified as having previously communicated with the user identified by the user identifier, the request comprising the user identifier and the application identifier.
45. The computer program product of claim 39, wherein the authentication server sends a request to a second client identified as having previously corresponded with the user identified by the user identifier, the request comprising the user identifier and the application identifier.
46. The computer program product of claim 39, wherein the authentication server sends a plurality of requests to a plurality of clients identified as having previously interacted with the user identified by the user identifier, the requests comprising the user identifier and the application identifier.
47. The computer program product of claim 39, wherein the authentication server receives a plurality of responses from plurality of clients, wherein at least one of the responses confirms the user identifier corresponds to the application identifier.
48. The computer program product of claim 39, wherein the authentication server: receives a plurality of responses from plurality of clients, wherein at least one of the responses confirms the user identifier corresponds to the application identifier; and determines that the number of responses confirming that the user identifier corresponds to the client address exceeds a given threshold
49. The computer program product of claim 39, wherein the authentication server sends, to a second server, a request to deduct a given amount from an account associated with the user identifier.

50. A method of providing targeted advertisements to users viewing media files in a distributed network, the method comprising:
- (a) receiving, by a server from a user operating a client, a request to access a media file, the request comprising a user identifier identifying the user and a media file identifier identifying the media file;
 - (b) determining, in response to the media file identifier and the user identifier, an advertisement to be displayed to the user in conjunction with the media file; and
 - (c) transmitting, to the client, the determined advertisement and a session key corresponding to the media file.
51. The method of claim 50, wherein step (a) comprises receiving, by a server from a first client, a request to access a media file, the request comprising a user identifier, a media file identifier, and an application identifier identifying an application requesting access to the media file.
52. The method of claim 51, wherein step (b) comprises determining, in response to the media file identifier, the user identifier, and the application identifier, an advertisement to be displayed to the user in conjunction with the media file.
53. The method of claim 50, wherein step (b) comprises determining, based on at least one media file previously accessed by the user, an advertisement to be displayed to the user in conjunction with the media file.
54. The method of claim 50, wherein step (b) comprises determining, based on at least one advertisement previously accessed by the user, an advertisement to be displayed to the user in conjunction with the media file.
55. The method of claim 50, wherein step (b) comprises determining, based on a length of time a previous advertisement was played by the user, an advertisement to be displayed to the user in conjunction with the media file.

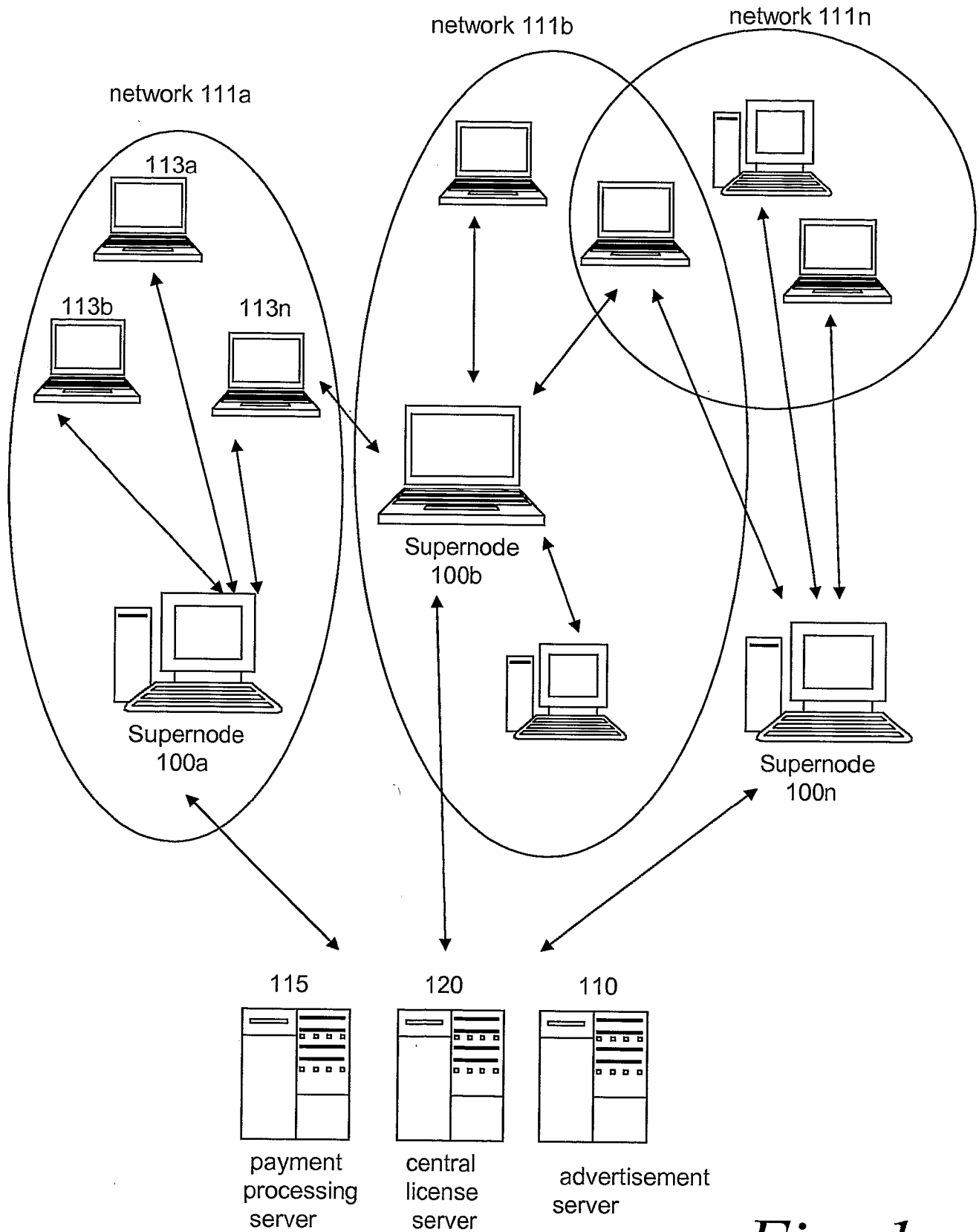
56. The method of claim 50, wherein step (b) comprises determining, based on a response to a previous advertisement by the a user corresponding to the user, an advertisement to be displayed to the user in conjunction with the media file.
57. The method of claim 50, wherein step (b) comprises determining, based on a purchase corresponding to a previous advertisement by the user, an advertisement to be displayed to the user in conjunction with the media file.
58. The method of claim 50, wherein step (b) comprises determining, based on at least one advertisement previously displayed in conjunction with the media file, an advertisement to be displayed to the user in conjunction with the media file.
59. The method of claim 50, wherein step (b) comprises determining, based on a length of time a previous advertisement was played by at least one other user also requesting the media file, an advertisement to be displayed to the user in conjunction with the media file.
60. The method of claim 50 wherein step (c) comprises transmitting, to the first client, a link to the determined advertisement and a session key corresponding to the media file.
61. The method of claim 50, wherein step (b) comprises determining, based on a response to a previous advertisement by at least one other user also requesting the media file, an advertisement to be displayed to the user in conjunction with the media file.
62. The method of claim 50, wherein step (b) comprises determining, based on a purchase corresponding to a previous advertisement by at least one other user also requesting the media file, an advertisement to be displayed to the user in conjunction with the media file.
63. A computer implemented system for providing targeted advertisements to users viewing media files in a distributed network, the system comprising

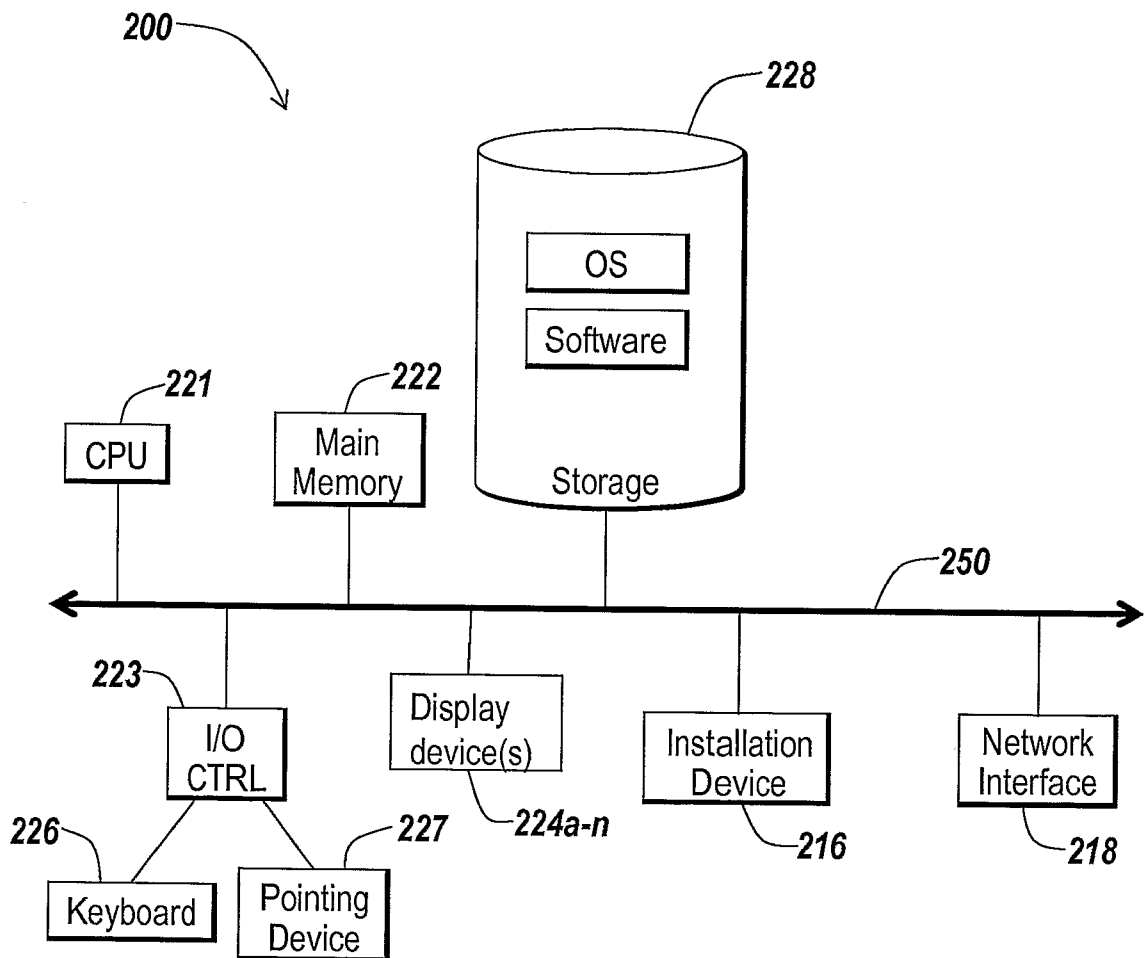
an authentication server which receives from a user operating a first client, a request to access a media file, the request comprising a user identifier identifying the user and a media file identifier identifying the media file; transmits the media file identifier and user identifier to an advertisement sever; and transmits to the client a session key corresponding to the media file and

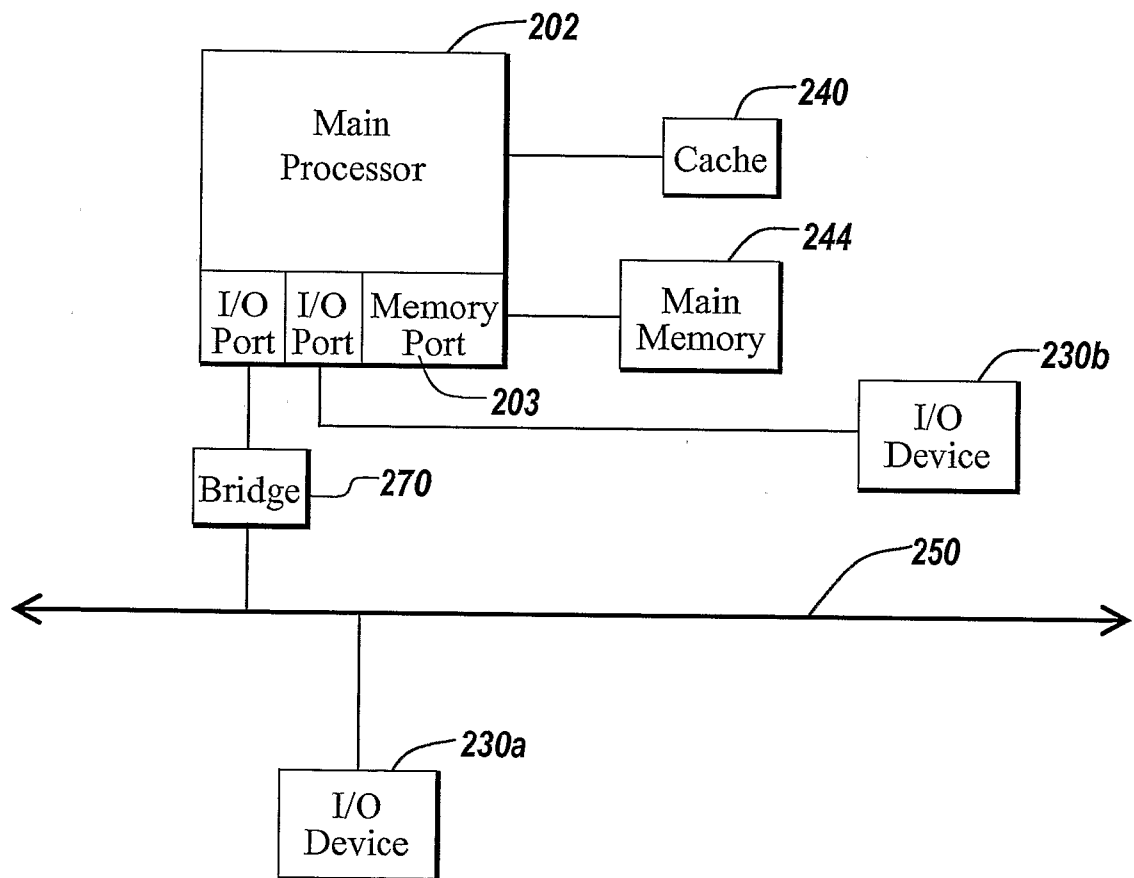
an advertisement server, in communication with the authentication server, which determines, in response to a received media file identifier and the user identifier, an advertisement to be displayed to the user in conjunction with the media file; and transmits, to the first client, the determined advertisement.

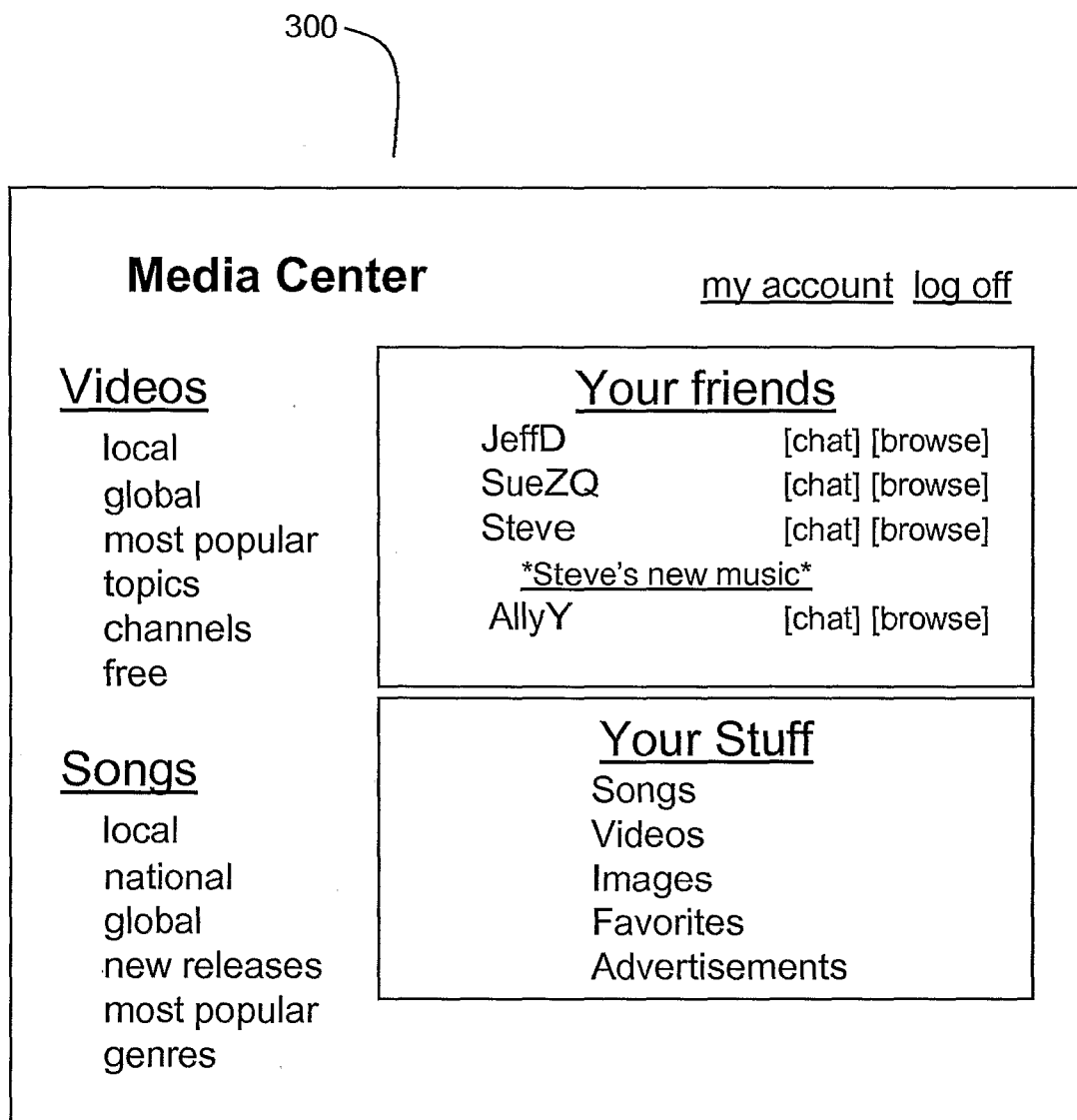
64. The system of claim 63, wherein the authentication server receives from a first client, a request to access a media file, the request comprising a user identifier, a media file identifier, and an application identifier identifying an application requesting access to the media file.
65. The system of claim 64, wherein the advertisement server determines, in response to the media file identifier, the user identifier, and the application identifier, an advertisement to be displayed to the user in conjunction with the media file.
66. The system of claim 63, wherein the advertisement server determines, based on at least one media file previously accessed by the user, an advertisement to be displayed to the user in conjunction with the media file.
67. The system of claim 63, wherein the advertisement server determines, based on at least one advertisement previously accessed by the user, an advertisement to be displayed to the user in conjunction with the media file.
68. The system of claim 63, wherein the advertisement server determines, based on a length of time a previous advertisement was played by the user, an advertisement to be displayed to the user in conjunction with the media file.

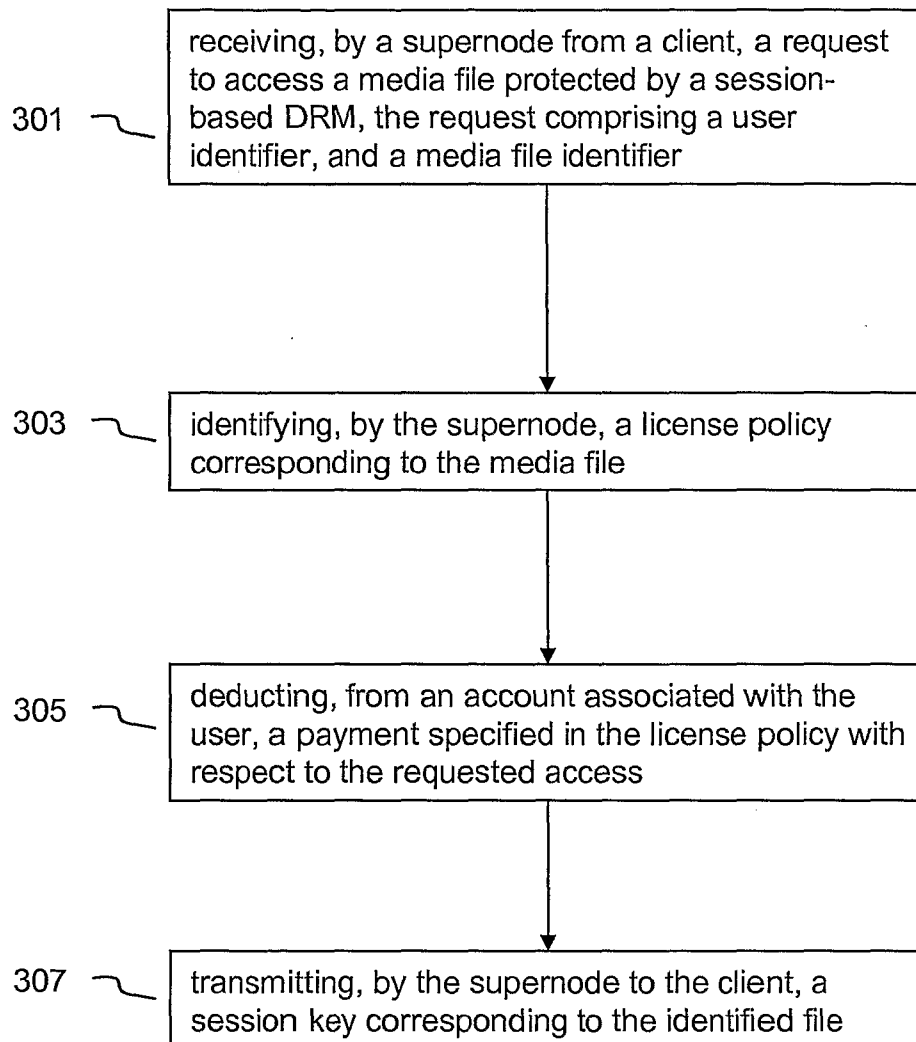
69. The system of claim 63, wherein the advertisement server determines, based on a response to a previous advertisement by the a user corresponding to the user, an advertisement to be displayed to the user in conjunction with the media file.
70. The system of claim 63, wherein the advertisement server determines, based on a purchase corresponding to a previous advertisement by the user, an advertisement to be displayed to the user in conjunction with the media file.
71. The system of claim 63, wherein the advertisement server determines, based on at least one advertisement previously displayed in conjunction with the media file, an advertisement to be displayed to the user in conjunction with the media file.
72. The system of claim 63, wherein the advertisement server determines, based on a length of time a previous advertisement was played by at least one other user also requesting the media file, an advertisement to be displayed to the user in conjunction with the media file.
73. The system of claim 63, wherein the advertisement server determines, based on a response to a previous advertisement by at least one other user also requesting the media file, an advertisement to be displayed to the user in conjunction with the media file.
74. The system of claim 63, wherein the advertisement server determines, based on a purchase corresponding to a previous advertisement by at least one other user also requesting the media file, an advertisement to be displayed to the user in conjunction with the media file.
75. The system of claim 63, wherein the authentication server transmits, to the first client, a link to the determined advertisement and a session key corresponding to the media file.

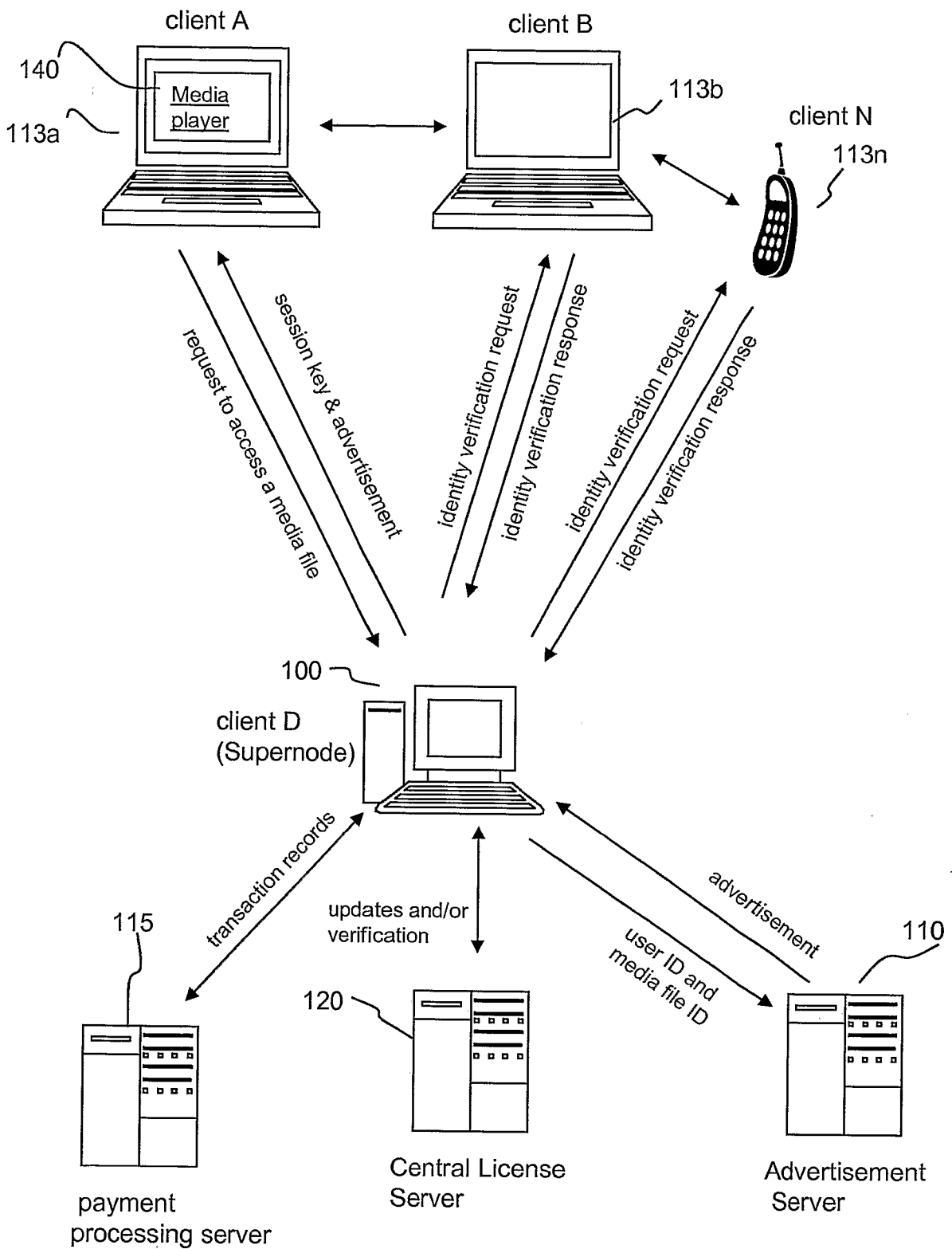
*Fig. 1*

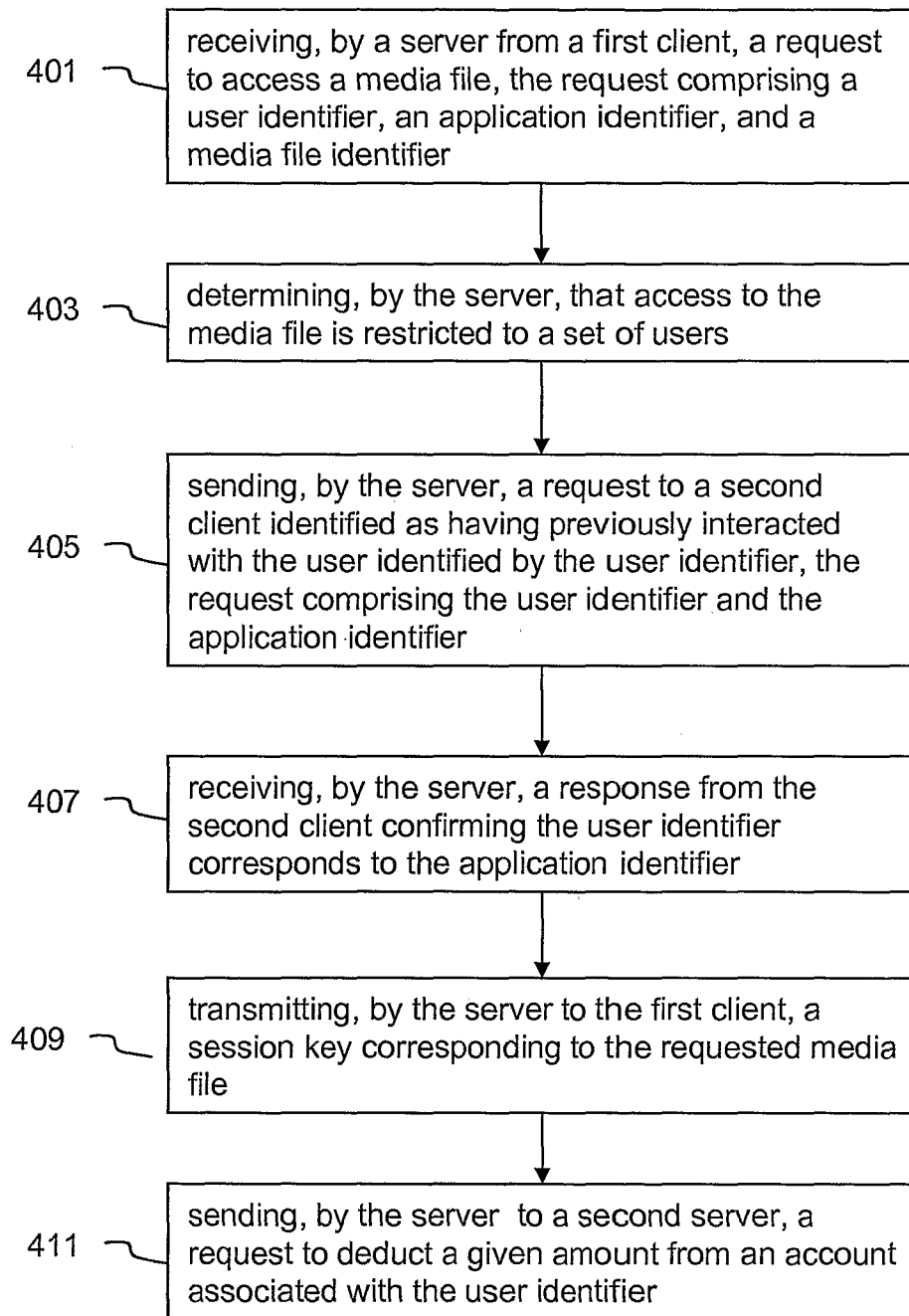
*Fig. 2A*

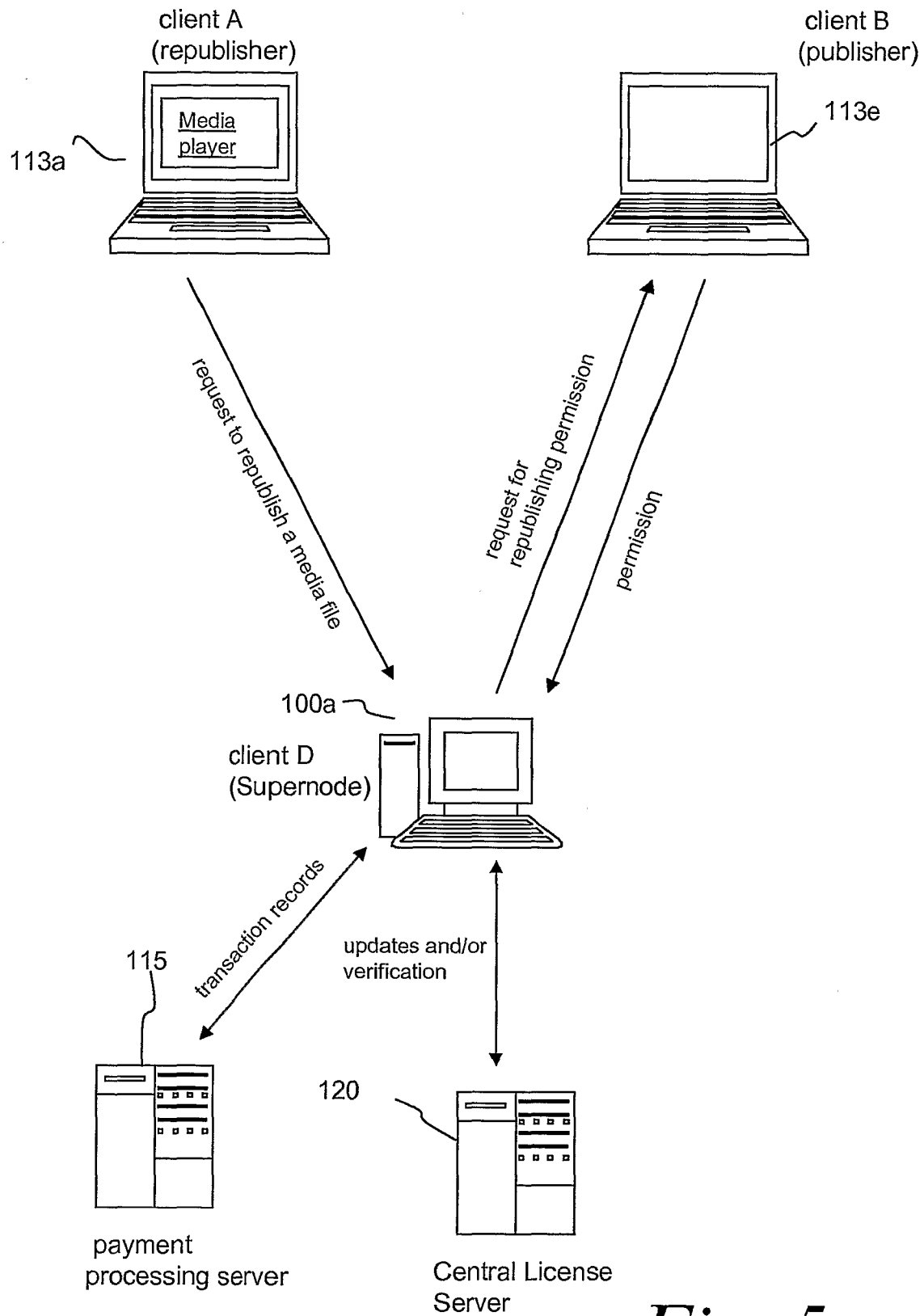
*Fig. 2B*

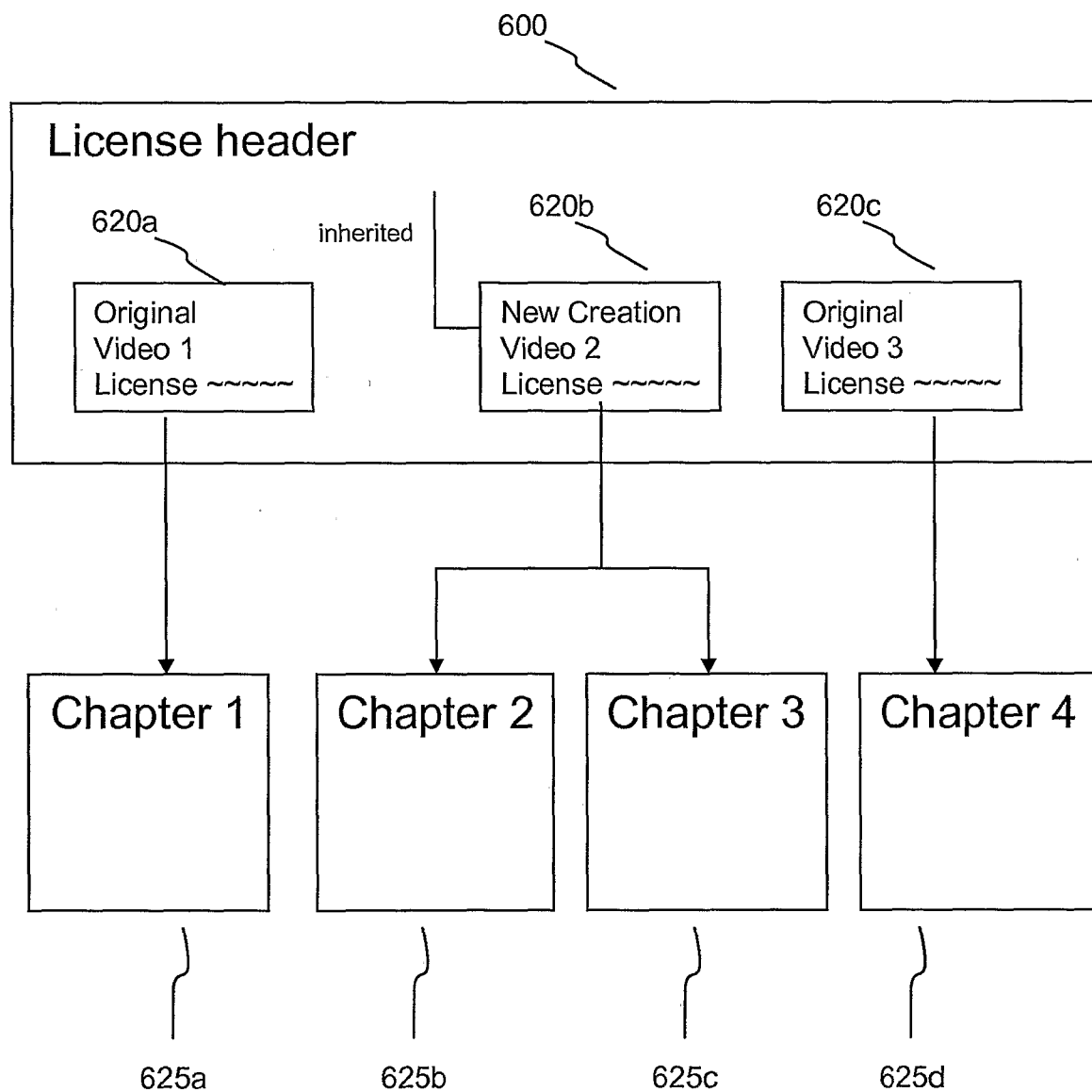
*Fig. 3A*

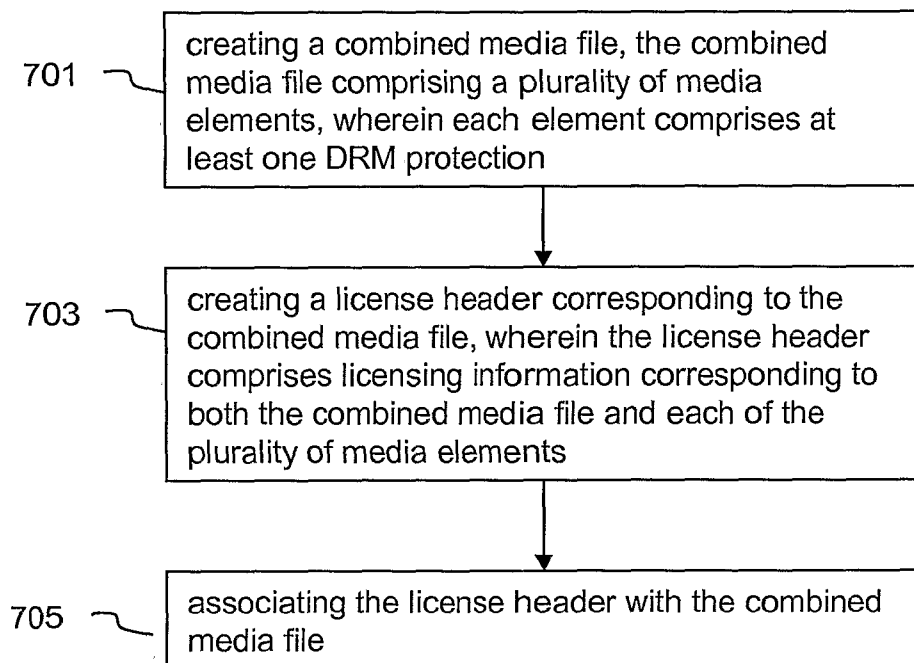
*Fig. 3B*

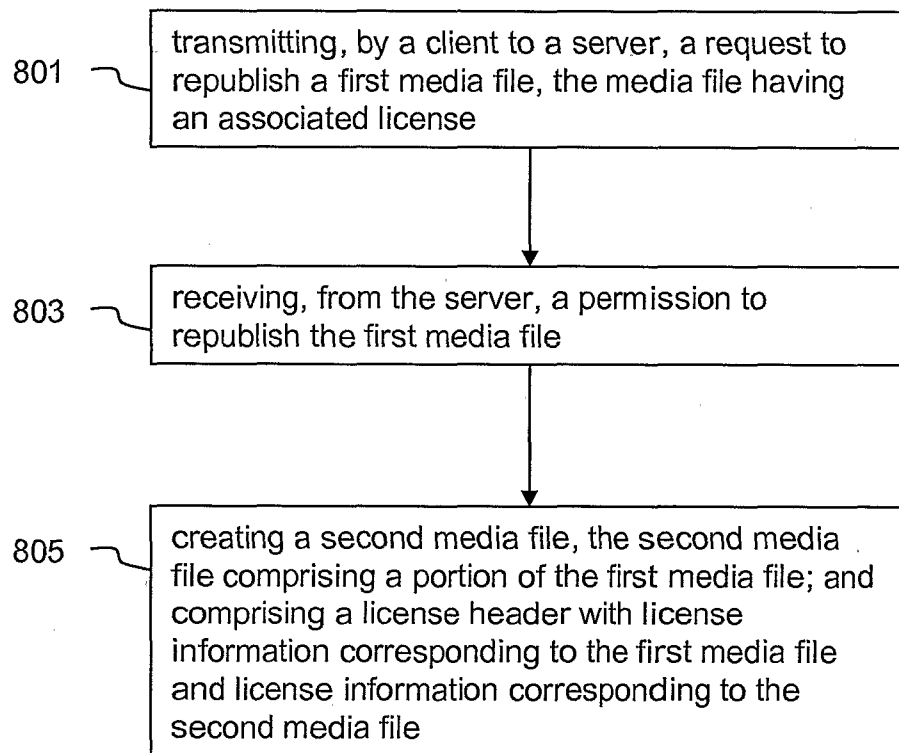
*Fig. 4A*

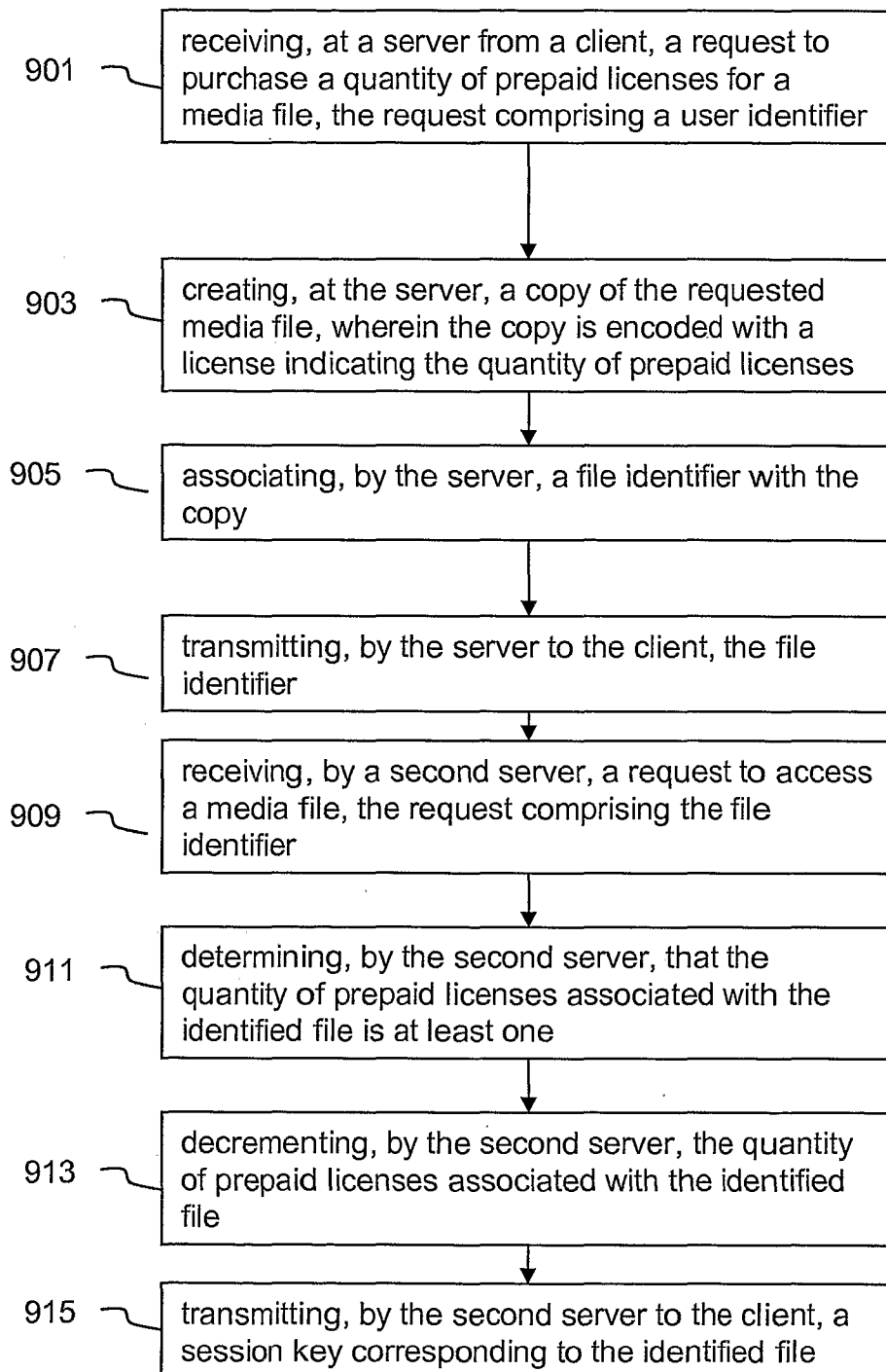
*Fig. 4B*

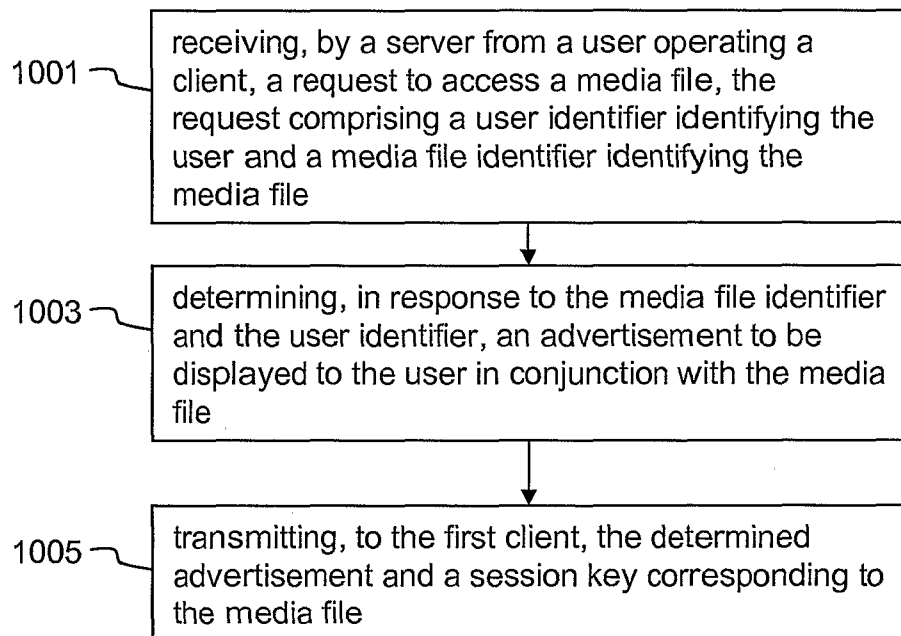
*Fig. 5*

*Fig. 6*

*Fig. 7*

*Fig. 8*

*Fig. 9*

*Fig. 10*

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2006/060999

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2006/048804 A (KONINKL PHILIPS ELECTRONICS NV [NL]; VRIELINK KOEN H J [NL]; LENOIR PE) 11 May 2006 (2006-05-11) pages 6-7	1-49
A	WO 01/80525 A (SUN MICROSYSTEMS INC [US]) 25 October 2001 (2001-10-25) abstract; figures 1,6	1-49
A	WO 03/034408 A (NOKIA CORP [FI]; NOKIA INC [US]) 24 April 2003 (2003-04-24) abstract; figure 5 claims 37-39	1-49



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

A document defining the general state of the art which is not considered to be of particular relevance

E earlier document but published on or after the international filing date

L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

O document referring to an oral disclosure, use, exhibition or other means

P document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

& document member of the same patent family

Date of the actual completion of the international search

11 October 2007

Date of mailing of the international search report

06/02/2008

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

KERSCHBAUMER, J

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2006/060999

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers allsearchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search reportcovers only those claims for which fees were paid, specifically claims Nos.:
4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-49

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☐ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-49

A method for providing collaborative authentication

2. claims: 50-75

A method of providing targeted advertisements

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2006/060999

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
WO 2006048804	A	11-05-2006	CN 101053235 A KR 20070085682 A	10-10-2007 27-08-2007
WO 0180525	A	25-10-2001	AU 4529201 A	30-10-2001
WO 03034408	A	24-04-2003	AU 2002362952 A1 CN 1650559 A EP 1444690 A2 US 2003076955 A1	28-04-2003 03-08-2005 11-08-2004 24-04-2003