A peer to peer software license management system that is capable of temporarily relocating available software licenses is disclosed. Any node (100, 200, 300) of the peer to peer software license management system (i.e., any computer in which the software license management system software of the invention is installed) is allowed to act as a license provider, or act as a license consumer, or act as both a license provider and a license consumer. Any node of the peer to peer software license management system has the capacity to temporarily borrow a software license from another node, and to re-serve the borrowed software license to yet another node as if the node that borrowed the software license were itself a license provider. Each node of the peer to peer software license management system operates as both a license provider and a license consumer.
before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))
PEER TO PEER SOFTWARE LICENSE MANAGEMENT SYSTEM FOR TEMPORARILY RELOCATING AVAILABLE SOFTWARE LICENSES

RELATED APPLICATION

[0001] This patent application is related to another patent application by the same inventors entitled "Software License Management System That Functions In A Disconnected Or Intermittently Connected Mode" filed on even date herewith as United States Patent Application Serial No. [Docket No. H0017806-0104]. The related patent application is assigned to the assignee of the present patent application. The disclosure of the related patent application is hereby incorporated by reference into the present application as if fully set forth herein.

TECHNICAL FIELD

[0002] This disclosure relates generally to software license management systems and more specifically to a peer to peer software license management system for temporarily relocating available software licenses.
Software license management is an essential service for any commercial software product to ensure the legitimate usage of the software product and to ensure the proper distribution of the software product. Software license providers ("license providers") are typically located within software that is located on a computer server. Software license consumers ("license consumers") are typically software applications that are located on a computer workstation or computer server. Existing software license management systems are based upon the ability of the software license provider to communicate securely with the software license consumers.

A license provider has a finite number of software licenses ("licenses") stored within the computer server on which the license provider is located. The license provider typically has an encrypted license file that contains the names of the licenses and the names of the customers (i.e., license consumers) that are authorized to receive the licenses.

When a license consumer desires to obtain particular software licenses, the license consumer may attempt to acquire the licenses from one or more license providers. The license consumer sends a license request to one or more license providers. A license provider that can satisfy the license consumer's request (1) licenses the requested license (or licenses) to the license consumer, (2) marks the requested license (or licenses) as being in use by the license consumer, and (3) establishes and maintains a secure communication link with the license consumer.

The license provider maintains the secure
communication link with the license consumer until the license consumer releases the license (or licenses) or the link with the license consumer fails due to a hardware or software fault, thereby implicitly releasing the license (or licenses). A license that is marked as being in use is not available to other license consumers until the license has been released. Licenses that are released re-enter the available license pool of the license provider.

[0007] The encrypted license file in the software license management system may be stored on a disk drive or other memory device such as a Universal Serial Bus (USB) key or smartcard. As previously mentioned, the encrypted license file contains the names of the authorized license consumers and the names of the licenses that they are authorized to receive. The software license management system is capable of reading the authorization information in the encrypted license file and serving the requested licenses to the license consumers that have the proper authorization.

[0008] The software license management system typically resides at a location that is remote from the location of the software applications that it licenses. This provides a more convenient centralized administration of the license process. In existing software license management systems, the license provider (located on a license server) is located at a remote location with respect to the license consumer (located on a client server).

[0009] Dedicated license servers are not efficient in mobile computing scenarios. For example, dedicated license servers are not efficient at supporting software product demonstrations or training operations at remote non-fixed locations. Prior art approaches to providing software licenses in mobile computing scenarios include providing
(for an additional fee) (1) short-term or dedicated entitlement to support mobile use, (2) portable entitlement in the form of hardware keys, (3) temporary offline stand-alone licenses, and (4) node-locked licenses that are not network-based.

[0010] It would be desirable to have a more efficient method for providing software license entitlement in mobile computing scenarios. It would be desirable to have a software license management system that has the ability to temporarily relocate available software licenses in mobile computing scenarios.
The present invention provides a peer to peer software license management system that is capable of temporarily relocating available software licenses. The present invention comprises a peer to peer software license management system in which any node (i.e., any computer in which the software license management system software of the invention is installed) is allowed to act as a license provider, or act as a license consumer, or act as both a license provider and a license consumer.

Any node of the peer to peer software license management system of the invention has the capacity (1) to temporarily borrow a software license from another node, and (2) re-serve the borrowed software license to yet another node as if the node that borrowed the software license were itself a license provider. Each node of the peer to peer software license management system of the invention operates as both a license consumer and a license provider.

Other technical features may be readily apparent to one skilled in the art from the following figures, descriptions, and claims.
BRIEF DESCRIPTION OF THE DRAWINGS

[0014] For a more complete understanding of this disclosure, reference is now made to the following description, taken in conjunction with the accompanying drawings, in which:

[0015] FIGURE 1 illustrates a schematic representation of an exemplary first computer node of the peer to peer software license management system of the invention;

[0016] FIGURE 2 illustrates a schematic representation of a first computer node of the invention accessing a software license from a second computer node of the invention;

[0017] FIGURE 3 illustrates a schematic representation of a first computer node of the invention temporarily borrowing a software license from a second computer node of the invention;

[0018] FIGURE 4 illustrates a schematic representation of a third computer node of the invention accessing a temporarily borrowed software license from the first computer node of the invention;

[0019] FIGURE 5 illustrates a flow chart showing the steps of an advantageous embodiment of a method of the present invention; and

[0020] FIGURE 6 illustrates a flow chart showing the steps of another advantageous embodiment of a method of the present invention.
FIGURES 1 through 6 and the various embodiments used to describe the principles of the present invention in this patent document are by way of illustration only and should not be construed in any way to limit the scope of the invention. Those skilled in the art will understand that the principles of the invention may be implemented in any type of suitably arranged device or system.

Every computer that participates in the peer to peer software license management system of the invention is referred to as a node. In a peer to peer system, each node can accept incoming connections from other nodes and each node can initiate outgoing connections to other nodes. This feature distinguishes a peer to peer system from a client-server system. In a client-server system, only client nodes can initiate outgoing connections and only server nodes can accept incoming connections.

FIGURE 1 illustrates a schematic representation of an exemplary first computer node 100 (also designated as node 1) of the peer to peer software license management system of the invention. First computer node 100 is representative of each of the other nodes in the system (not shown in FIGURE 1). As shown in FIGURE 1, first computer node 100 comprises an application process 110. Application process 110 comprises at least one program 120 and a licensing library 130. The licensing library 130 contains one or more software licenses. The program 120 is capable of accessing the licensing library 130 through a licensing application programming interface (designated as "Licensing API" in FIGURE 1).

First computer node 100 also comprises a continuously running program that is referred to as a
license service 140 or a license daemon 140. The license daemon 140 is connected to and in communication with the licensing library 130. The licensing library 130 is capable of accessing the license daemon 140 to pass on to the license daemon 140 any license requests that are made by the program 120.

[0025] First computer node 100 also comprises an encrypted license file 150. The encrypted license file 150 contains the names of authorized license consumers and the names of the licenses that they are authorized to receive. The license daemon 140 is connected to and in communication with the encrypted license file 150. The license daemon 140 is capable of reading the authorization information in the encrypted license file 150 and serving the requested licenses to the license consumers that have the proper authorization.

[0026] First computer node 100 also comprises an address file 160 that contains computer addresses of remotely located licensing nodes in the peer to peer system. The license daemon 140 is connected to and in communication with the address file 160. The license daemon 140 is capable of communicating with the remotely located nodes in the peer to peer system.

[0027] The standard computer hardware of computer node 100 (central processing unit, memory units, etc.) and the standard computer software of computer node 100 (operating system, etc.) together with the software of the peer to peer software license management system of the invention (license daemon, license file, licensing library, etc.) comprises a software license management controller that performs the functions of the present invention.

[0028] FIGURE 2 illustrates a schematic representation of the first computer node 100 accessing a software license
from a second computer node 200 of the peer to peer system of the invention. Second computer node 200 comprises an application process 210 that comprises at least one program 220 and a licensing library 230. Second computer node 200 also comprises its own license daemon 240 and encrypted license file 250. Although not shown in FIGURE 2, the second computer node 200 also comprises an address file that contains computer addresses of remotely located licensing nodes in the peer to peer system.

[0029] FIGURE 2 illustrates how the first computer node 100 may access a software license from the second computer node 200 in the peer to peer system. When the at least one program 120 makes a license request, the licensing library 130 sends the request to the license daemon 140. The license daemon 140 first attempts to access (or to "check out") the requested license from its own encrypted license file 150 (not shown in FIGURE 2). If the requested license is in the encrypted license file 150, then the license daemon 140 provides the requested license to the program 120 from the licensing library 130.

[0030] If the requested license is not in the encrypted license file 150 (i.e., the requested license is not locally available), then the license daemon 140 passes the license request to the known remote nodes (which are also referred to as remote hosts). FIGURE 2 illustrates the connection and communication between license daemon 140 of the first computer node 100 and the license daemon 240 of the second computer node 200.

[0031] The license daemon 240 accesses its own encrypted license file 250 to determine whether the requested license is located within the second computer node 200. If the requested license is located within the encrypted license file 250, then the license daemon 240
marks the requested license as being in use and grants the license request. The license daemon 240 provides the requested license to the license daemon 140 from the licensing library 230. The license daemon 140 provides the requested license to the licensing library 130 and to the requesting program 120.

[0032] If the requested license is not located within the encrypted license file 250 in the second computer node 200, then the license daemon 240 reports that finding back to the license daemon 140 in the first computer node 100.

[0033] When the license daemon 140 sends out a license request to the other nodes in the peer to peer network, the other nodes will cascade the request to other nodes in their local address list (if any) until the requested license is found or the license search fails due the lack of availability of the requested license on any node that is connected in the peer to peer network.

[0034] In the search for a requested license, a success or a failure will be reported back to the requesting program 120 via the API link. A successfully accessed license will eventually be returned to the remote computer node from which the license came (1) when the requesting program 120 releases the license via the API link or (2) when the requesting program 120 terminates. Returning the license to the remote computer node from which the license came requires the local license daemon 140 to track where the license originated and to communicate with the license daemon on the remote computer node (e.g., license daemon 240 in our present example) to inform the license daemon on the remote computer node of the eventual release of the requested license. After the requested license has been released, the license daemon on
the remote computer node that provided the requested license from its local encrypted license file releases the license back into the local pool of licenses that are eligible for a subsequent license.

[0035] In order prevent abuse of license entitlement, the license daemon of the computer node that provides the requested license generates and sends an occasional "all is well" signal (referred to as a "heartbeat") to all of the other computer nodes that are participating in a license access (also referred to as a license "checkout"). If for some reason the heartbeat signal is not acknowledged as expected, then the heartbeat signal is said to have failed. If the heartbeat signal fails, then all license daemons that are involved in the license checkout process immediately consider that the license checkout process has failed.

[0036] The failure of the heartbeat signal results in the requesting program (e.g., program 120) being informed via the API link that a license that it had successfully requested is now unavailable and that the previously successful license request has been terminated. The license daemon that provided the requested license from its local encrypted license file then releases the license back into the local pool of licenses that are eligible for a subsequent license. No heartbeat signal is required when a license request is satisfied from a local license file.

[0037] In addition to being checked out, licenses can also be borrowed. Borrowing a license entails relocating a license from one license file to another license file for a predetermined (but finite) period of time.

[0038] A license daemon in a computer node can request to borrow one or more licenses from another computer node. The license borrowing process is conceptually similar to a
normal license check out process from one node to another. But there is no heartbeat signal maintained between the two nodes that are involved in the borrowing transaction. This is so that the borrowing node can be disconnected from the network of the donor node and continue to use the borrowed license while the borrowing node is offline (i.e., not connected to the peer to peer network). The donor node considers that the borrowed license has been checked out. But the borrowing node considers the borrowed license to be available for check out by programs that are running locally on the borrowing node.

[0039] FIGURE 3 illustrates how the first computer node 100 may temporarily borrow a software license from the second computer node 200 in the peer to peer system. When the at least one program 120 makes a license request, the licensing library 130 sends the request to the license daemon 140. The license daemon 140 first attempts to access (or to "check out") the requested license from its own encrypted license file 150 (not shown in FIGURE 3). If the requested license is in the encrypted license file 150, then the license daemon 140 provides the requested license to the program 120 from the licensing library 130.

[0040] If the requested license is not in the encrypted license file 150 (i.e., the requested license is not locally available), then the license daemon 140 may issue a request to borrow the license from one of the known remote nodes (which are also referred to as remote hosts). FIGURE 3 illustrates the connection and communication between license daemon 140 of the first computer node 100 and the license daemon 240 of the second computer node 200. Although not shown in FIGURE 3, the second computer node 200 comprises a previously mentioned address file that contains computer addresses of remotely located licensing
nodes in the peer to peer system.

[0041] The license daemon 240 accesses a donor license file 270 to determine whether the requested license is a license that is authorized to be "loaned" out as a borrowed license. If the requested license is located within the donor license file 270, then the license daemon 240 marks the requested license as being in use for the requested time period and grants the license request. The license daemon 240 provides the requested borrowed license to the license daemon 140 from the licensing library 230. The license daemon 140 stores the requested borrowed license in a borrowed licenses file 170 and allows the requesting program 120 to access the requested borrowed license.

[0042] If the requested borrowed license is not located within the donor license file 270, then the license daemon 240 reports that finding back to the license daemon 140 in the first computer node 100.

[0043] When the license daemon 140 sends out a request to borrow a license to the other nodes in the peer to peer network, the other nodes will cascade the request to other nodes in their local address list (if any) until the requested license is found or the license search fails due the lack of availability of the requested license on any node that is connected in the peer to peer network.

[0044] As previously mentioned, when the license daemon 240 sends the borrowed license to the license daemon 140, the license daemon 140 stores the borrowed license in the borrowed licenses file 170. The borrowed licenses that are stored in the borrowed licenses file 170 in the first computer node 100 appear as normal local licenses during the requested time period (i.e., the "lifetime" of the borrowed license) .
After the borrowed license is successfully stored in the borrowed licenses file 170 in the first computer node 100, the network connection between the license daemon 140 of first computer node 100 and the license daemon 240 of the second computer node 200 is severed. The severance is schematically represented in FIGURE 3 with the scissors symbol 290. The borrowing node (first computer node 100 in our present example) may continue to use the borrowed license while offline (i.e., not connected to the peer to peer network).

The borrowing node (first computer node 100 in our present example) may check the borrowed license back into the donor node (second computer node 200 in our present example) at any time the borrowing node and the donor node are connected on a common network. Checking the borrowed license back into the donor node ends the period of borrowing.

To secure the borrowing transaction and enforce the time limit for the borrowed license, it is necessary to specify a maximum borrowing time limit at the time when the borrowed license is borrowed. The maximum borrowing time limit (also referred to as the "timeout" value) is tracked independently by both the donor node and the borrowing node. If the timeout value expires before the borrowed license is returned to the donor node from the borrowing node, then the borrowed license will automatically be disabled on the borrower node and will automatically be enabled on the donor node (i.e., checked back into the donor node).

During the time that the borrowed license is still within the timeout value, the borrowed license remains valid. Furthermore, a valid borrowed license is considered to be available for checkout from the borrowing
node. Due to the peer to peer capabilities of the network, the borrowed license is not restricted to checkout only by programs that are running on the borrowing node. That is, the borrowed license may be served to other connecting nodes that request it. This means that a license can be temporarily relocated from a donor node (that acts as a license server) to a borrowing node (that subsequently goes offline with respect to the donor node) and the borrowing node can subsequently act as a license server for the borrowed license.

[0049] FIGURE 4 illustrates how the first computer node 100 may act as a license server for a borrowed license and serve the borrowed license to a third computer node 300 in the peer to peer system. Third computer node 300 comprises an application process 310 that further comprises at least one program 320 and a licensing library 330. Third computer node 300 also comprises its own license daemon 340 and address file 360. Although not shown in FIGURE 4, the third computer node 300 also comprises an encrypted license file.

[0050] When the at least one program 320 makes a license request, the licensing library 330 sends the request to the license daemon 340. The license daemon 340 first attempts to access (or to "check out") the requested license from its own encrypted license file (not shown in FIGURE 4). If the requested license is in the encrypted license file, then the license daemon 340 provides the requested license to the program 320 from the licensing library 330.

[0051] If the requested license is not in the encrypted license file (i.e., the requested license is not locally available), then the license daemon 340 may issue a request to access (or "check out") the license from one of
the known remote nodes (which are also referred to as remote hosts). FIGURE 4 illustrates the connection and communication between license daemon 340 of the third computer node 300 and the license daemon 140 of the first computer node 100. Although not shown in FIGURE 4, the first computer node 100 comprises an encrypted license file 150 and an address file 160 that contains computer addresses of remotely located licensing nodes in the peer to peer system.

[0052] The license daemon 140 accesses the borrowed licenses file 170 to determine whether the requested license is available. If the requested license is located within the borrowed licenses file 170, then the license daemon 140 marks the requested license as being in use and grants the license request. The license daemon 140 provides the requested license to the licensing daemon 340 from the borrowed licenses file 170. The license daemon 340 provides the borrowed license to the licensing library 330 and to the requesting program 320.

[0053] If the requested license is not located within the borrowed licenses file 170, then the license daemon 140 reports that finding back to the license daemon 340 in the third computer node 300.

[0054] When the license daemon 340 sends out a request to access a license to the other nodes in the peer to peer network, the other nodes will cascade the request to other nodes in their local address list (if any) until the requested license is found or the license search fails due the lack of availability of the requested license on any node that is connected in the peer to peer network.

[0055] As previously mentioned, when the license daemon 140 sends the borrowed license to the license daemon 340, the license daemon 340 provides the borrowed license
The first computer node 100 is a borrowing node that has gone offline with respect to the original donor node (second computer node 200). In this manner, the first computer node 100 acts in turn as a license server that serves the previously borrowed license to the third computer node 300.

[0056] FIGURE 5 illustrates a flow chart 500 that shows the steps of an advantageous embodiment of a method of the present invention. In the first step, a peer to peer network of computer licensing nodes is provided (step 510). Then a requesting program 120 in a first computer node 100 issues a license request to its license daemon 140 (step 520). The license daemon 140 searches for the requested license in a local license file 150 and in a local licensing library 130 and, if found, sends the requested license to the requesting program 120 (step 530).

[0057] If the license daemon 140 does not find the requested license in the first computer node 100 then the license daemon 140 sends the license request to each computer node in the peer to peer network for which the license daemon 140 has an address and the license request cascades through the network nodes (step 540). Then a license daemon 240 in a second computer node 200 determines that the requested license is in its local license file 250 and is in its local licensing library 230 (step 550).

[0058] Then the license daemon 240 (1) marks the requested license as being in use and (2) grants the license request to license daemon 140 and (3) sends out a heartbeat signal to license daemon 140 for the duration of the license grant (step 560). The license terminates when (1) requesting program 120 releases the license or (2) when requesting program 120 terminates or (3) when the heartbeat signal fails (step 570). When the license...
terminates then license daemon 240 releases the license back into the local pool of licenses that are eligible for a subsequent license (step 580).

[0059] FIGURE 6 illustrates a flow chart 600 that shows the steps of another advantageous embodiment of a method of the present invention. In the first step, a peer to peer network of computer licensing nodes is provided (step 610). Then a requesting program 120 in a first computer node 100 issues a license request to its license daemon 140 (step 620). The license daemon 140 searches for the requested license in a local license file 150 and in a local licensing library 130 and, if found, sends the requested license to the requesting program 120 (step 630).

[0060] If the license daemon 140 does not find the requested license in the first computer node 100 then the license daemon 140 sends a borrow request to each computer node in the peer to peer network for which the license daemon 140 has an address and the borrow request cascades through the network nodes (step 640). Then a license daemon 240 in a second computer node 200 determines that the requested license is in its local donor file 270 and is in its local licensing library 230 (step 650).

[0061] Then the license daemon 240 (1) marks the requested license as being in use for a requested time limit and (2) sends the requested borrowed license to license daemon 140 and to the borrowed licenses file 270 in the first computer node 100 (step 660). The network connection between the first computer node 100 and the second computer node 200 is then severed and the first computer node 100 operates offline with the borrowed license (step 670).

[0062] While operating offline the first computer node 100 then licenses the borrowed license to a third computer...
node 300 (step 680). The borrowed license automatically terminates when the original requested time limit expires (step 690). When the borrowed license terminates then license daemon 240 releases the borrowed license back into the local pool of licenses that are eligible for a subsequent license or a subsequent borrowing (step 695).

[0063] It may be advantageous to set forth definitions of certain words and phrases used throughout this patent document. The terms "application," "program," and "routine" refer to one or more computer programs, sets of instructions, procedures, functions, objects, classes, instances, or related data adapted for implementation in a suitable computer language. The term "couple" and its derivatives refer to any direct or indirect communication between two or more elements, whether or not those elements are in physical contact with one another.

[0064] The terms "transmit," "receive," and "communicate," as well as derivatives thereof, encompass both direct and indirect communication. The terms "include" and "comprise," as well as derivatives thereof, mean inclusion without limitation. The term "or" is inclusive, meaning and/or. The phrases "associated with" and "associated therewith," as well as derivatives thereof, may mean to include, be included within, interconnect with, contain, be contained within, connect to or with, couple to or with, be communicable with, cooperate with, interleave, juxtapose, be proximate to, be bound to or with, have, have a property of, or the like. The term "controller" means any device, system, or part thereof that controls at least one operation. A controller may be implemented in hardware, firmware, software, or some combination of at least two of the same. The functionality associated with any particular controller may be centralized or distributed, whether
locally or remotely.

[0065] While this disclosure has described certain embodiments and generally associated methods, alterations and permutations of these embodiments and methods will be apparent to those skilled in the art. Accordingly, the above description of example embodiments does not define or constrain this disclosure. Other changes, substitutions, and alterations are also possible without departing from the spirit and scope of the invention, as defined by the following claims.
WHAT I S CLAIMED IS:

1. An apparatus for managing software licenses comprising:
   a plurality of computers (100, 200, 300) connected together in a peer to peer computer network wherein each computer is a computer node of the network;
   wherein each computer node comprises a software license management controller that operates as a software license provider and a software license consumer.

2. The apparatus as set forth in Claim 1 wherein the software license management controller in each computer node comprises a license daemon (140) that is in communication with other license daemons (140) in other computer nodes.

3. The apparatus as set forth in Claim 2 wherein the license daemon in a first computer node, in response to receiving a request for a software license from a requesting program in the first computer node, searches for the requested software license in the first computer node.

4. The apparatus as set forth in Claim 3 wherein the license daemon in the first computer node, in response to not finding the requested software license in the first computer node, sends to other license daemons in other computer nodes a request to borrow the software license.

5. The apparatus as set forth in Claim 4 wherein a license daemon in a second computer node determines that the requested software license is located in the second computer node; and marks the requested software license as
being in use for a requested time limit; and sends the requested software license to the license daemon in the first computer node.

6. The apparatus as set forth in Claim 5 wherein a connection of the first computer node that borrows a software license from the second computer node is severed from the second computer node after the first computer node has received the borrowed software license from the second computer node.

7. An apparatus for managing software licenses comprising:
   a plurality of computers (100, 200, 300) connected together in a peer to peer computer network wherein each computer is a computer node of the network;
   wherein each computer node comprises a software license management controller that comprises a license daemon (140) that is in communication with other license daemons (140) in other computer nodes;
   wherein a license daemon in a first computer node, in response to receiving a request for a software license from a requesting program in the first computer node, searches for the requested software license in the first computer node; and
   wherein, in response to not finding the requested software license in the first computer node, the license daemon in the first computer node sends to other license daemons in other computer nodes a request to license the software license.
8. The apparatus as set forth in Claim 7 wherein the requested software license terminates upon one of: the requesting program in the first computer node releases the requested software license, the requesting program in the first computer node terminates operation, and the heartbeat signal fails.

9. A method for managing software licenses comprising the steps of:
   connecting (510, 610) a plurality of computers in a peer to peer computer network wherein each computer is a computer node of the network:
   providing a software license management controller within each computer node; and
   operating (520-580, 620-695) the software license management controller as a software license provider and a software license consumer.

10. The method as set forth in Claim 9 further comprising the steps of:
    temporarily borrowing a software license from a software license management controller of a second computer node for use by a software license management controller of a first computer node; and
    granting a license from the software license management controller of the first computer node to a software license management controller of a third computer node to use the software license that was borrowed from the software license management controller of the second computer node.
FIG. 1
FIG. 2

COMPUTER (NODE 1)

APPLICATION PROCESS

PROGRAM REQUESTING LICENSE

LICENSING API

LICENSING LIBRARY

LICENSE DAEMON

120

130

140

160

LICENSE DAEMON

ADDRESSES OF REMOTE LICENSING NODES

(1) PROGRAM MAKES LICENSE REQUEST

(2) NO LOCAL LICENSES SO PASS REQUEST TO KNOWN REMOTE HOST(S)

COMPUTER (NODE 2)

APPLICATION PROCESS

PROGRAM REQUESTING LICENSE

LICENSING API

LICENSING LIBRARY

LICENSE FILE

LICENSE DAEMON

(3) LICENSE FILE HAS REQUESTED LICENSE

(4) DAEMON MARKS LICENSE 'IN USE' AND GRANTS THE REQUEST
FIG. 3

(1) REQUEST TO BORROW LICENSE FROM KNOWN REMOTE HOST

(2) DAEMONS MARKS LICENSE 'IN USE' FOR REQUESTED TIME PERIOD AND GRANTS THE REQUEST

(3) BORROWED LICENSE(S) APPEAR AS NORMAL LOCAL LICENSES FOR THE TIME PERIOD OF THE BORROWING

(4) NODE 1 IS THEN DISCONNECTED FROM THE DONOR NODE

COMPUTER (NODE 1)

APPLICATION PROCESS

PROGRAM REQUESTING LICENSE

LICENSING API

120

130

LICENSING LIBRARY

140

LICENSE DAEMON

BORROWED LICENSES

160

NODES

COMPUTER (NODE 2)

APPLICATION PROCESS

PROGRAM REQUESTING LICENSE

LICENSING API

220

230

LICENSING LIBRARY

200

DONOR LICENSE FILE

240

270
(1) PROGRAM MAKES LICENSE REQUEST

(2) NO LOCAL LICENSES SO PASS REQUEST TO KNOWN REMOTE HOST(S)

(3) BORROWED LICENSE AVAILABLE

(4) DAEMONS MARKS LICENSE 'IN USE' AND GRANTS THE REQUEST

FIG. 4
START

500 PROVIDE A PEER TO PEER NETWORK OF COMPUTER LICENSING NODES

520 REQUESTING PROGRAM 120 IN A FIRST COMPUTER NODE 100 ISSUES A LICENSE REQUEST TO ITS LICENSE DAEMON 140

530 LICENSE DAEMON 140 SEARCHES FOR THE REQUESTED LICENSE IN A LOCAL LICENSE FILE 105 AND IN A LOCAL LICENSING LIBRARY 130 AND, IF FOUND, SENDS THE REQUESTED LICENSE TO THE REQUESTING PROGRAM 120

540 IF THE LICENSE DAEMON 140 DOES NOT FIND THE REQUESTED LICENSE IN THE FIRST COMPUTER NODE 100 THEN THE LICENSE DAEMON 140 SENDS THE LICENSE REQUEST TO EACH COMPUTER NODE IN THE PEER TO PEER NETWORK FOR WHICH THE LICENSE DAEMON HAS AN ADDRESS AND THE LICENSE REQUEST CASCADES THROUGH THE NETWORK NODES

550 LICENSE DAEMON 240 IN A SECOND COMPUTER NODE 200 DETERMINES THAT THE REQUESTED LICENSE IS IN ITS LOCAL LICENSE FILE 250 AND IS IN ITS LOCAL LICENSING LIBRARY 230

560 LICENSE DAEMON 240 (1) MARKS THE REQUESTED LICENSE AS BEING IN USE AND (2) GRANTS THE LICENSE REQUEST TO LICENSE DAEMON 140 AND (3) SENDS OUT A HEARTBEAT SIGNAL TO LICENSE DAEMON 140 FOR THE DURATION OF THE LICENSE GRANT

570 THE LICENSE TERMINATES (1) WHEN REQUESTING PROGRAM 120 RELEASES THE LICENSE OR (2) WHEN REQUESTING PROGRAM 120 TERMINATES OR (3) WHEN THE HEARTBEAT SIGNAL FAILS

580 WHEN THE LICENSE TERMINATES THEN LICENSE DAEMON 240 RELEASES THE LICENSE BACK INTO THE LOCAL POOL OF LICENSES THAT ARE ELIGIBLE FOR A SUBSEQUENT LICENSE

CONTINUE

FIG. 5
600
START

610 PROVIDE A PEER TO PEER NETWORK OF COMPUTER LICENSING NODES

620 REQUESTING PROGRAM 120 IN A FIRST COMPUTER NODE 100 ISSUES A LICENSE REQUEST TO ITS LICENSE DAEMON 140

630 LICENSE DAEMON 140 SEARCHES FOR THE REQUESTED LICENSE IN A LOCAL LICENSE FILE 105 AND IN A LOCAL LICENSING LIBRARY 130 AND, IF FOUND, SENDS THE REQUESTED LICENSE TO THE REQUESTING PROGRAM 120

640 IF THE LICENSE DAEMON 140 DOES NOT FIND THE REQUESTED LICENSE IN THE FIRST COMPUTER NODE 100 THEN THE LICENSE DAEMON 140 SENDS A BORROW REQUEST TO EACH COMPUTER NODE IN THE PEER TO PEER NETWORK FOR WHICH THE LICENSE DAEMON HAS AN ADDRESS AND THE BORROW REQUEST CASCADES THROUGH THE NETWORK NODES

650 LICENSE DAEMON 240 IN A SECOND COMPUTER NODE 200 DETERMINES THAT THE REQUESTED LICENSE IS IN ITS LOCAL DONOR LICENSE FILE 270 AND IS IN ITS LOCAL LICENSING LIBRARY 230

660 LICENSE DAEMON 240 (1) MARKS THE REQUESTED LICENSE AS BEING IN USE FOR A REQUESTED TIME LIMIT AND (2) SENDS THE REQUESTED BORROWED LICENSE TO LICENSE DAEMON 140 AND TO THE BORROWED LICENSES FILE 170 IN THE FIRST COMPUTER NODE 100

670 THE NETWORK CONNECTION BETWEEN THE FIRST COMPUTER NODE 100 AND THE SECOND COMPUTER NODE 200 IS SEVERED AND THE FIRST COMPUTER NODE 100 OPERATES OFFLINE WITH THE BORROWED LICENSE

680 WHILE OPERATING OFFLINE THE FIRST COMPUTER NODE 100 THEN LICENSES THE BORROWED LICENSE TO A THIRD COMPUTER NODE 300

690 THE BORROWED LICENSE AUTOMATICALLY TERMINATES WHEN THE ORIGINAL REQUESTED TIME LIMIT EXPIRES

695 WHEN THE LICENSE TERMINATES THEN LICENSE DAEMON 240 RELEASES THE LICENSE BACK INTO THE LOCAL POOL OF LICENSES THAT ARE ELIGIBLE FOR A SUBSEQUENT LICENSE OR A SUBSEQUENT BORROWING

FIG. 6 CONTINUE
### A. CLASSIFICATION OF SUBJECT MATTER

**G06F 21/00(2006.01)1, G06F 21/20(2006.01)1, G06F 21/24(2006.01)1**

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)

<table>
<thead>
<tr>
<th>IPC8</th>
<th>G06F</th>
</tr>
</thead>
</table>

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

- Korean Utility models and applications for Utility models since 1975
- Japanese Utility models and applications for Utility models since 1975

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

- eKIPASS/KIPO internal) "program or firmware or software or application", "authentication or DRM or digital right or certification or license", "peer-to-peer or node-to-node or computer-to-computer or user-to-user", "borrow or lend or manage or control"

### C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No</th>
</tr>
</thead>
<tbody>
<tr>
<td>X A</td>
<td>JP 2007-2498044 A (KIMURA AKIRA et al) 27 September 2007 see abstract, pages 2-11, and figures 1-7</td>
<td>1-4, 7-10</td>
</tr>
<tr>
<td>A</td>
<td>US 2006-0143134 A1 (NICOL SO et al) 29 June 2006 see abstract, pages 2-7, and figures 1.7.1, 1.14</td>
<td>1-10</td>
</tr>
<tr>
<td>A</td>
<td>US 2007-0233782 A1 (TARIK TALJ) 4 October 2007 see abstract, pages 10-12, and figures 20-24</td>
<td>1-10</td>
</tr>
<tr>
<td>A</td>
<td>US 7,162,538 B1 (ROIS O CORDOBA) 9 January 2007 see abstract, column 1 line 45 - column 3 line 30, and figure 1</td>
<td>1-10</td>
</tr>
</tbody>
</table>

"X" 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 application or patent 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 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: 30 JULY 2009 (30 07 2009)

Date of mailing of the international search report: 31 JULY 2009 (31.07.2009)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 139 Seonsa-ro, Seogu, Daejeon 302-701, Republic of Korea

Facsimile No 82-42-472-7140

Authorized officer

LEE, OK WOO

Telephone No 82-42-481-5761

Form PCT/ISA/210 (second sheet) (July 2008)
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>JP 2007-249804 A</td>
<td>09.27.2007</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td>US 2006-143134 A1</td>
<td>29.06.2006</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td>US 2007-233782 A1</td>
<td>04.10.2007</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td>US 07162538 B1</td>
<td>09.01.2007</td>
<td>CA 2423722 A1 11.04.2002</td>
<td>EP 1415226 A2 06.05.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GB 2383869 A 09.07.2003</td>
<td>TW 594496 A 21.06.2004</td>
</tr>
</tbody>
</table>