US 20060190409A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2006/0190409 A1

Hillegass et al. (43) Pub. Date: Aug. 24, 2006

(54) METHOD AND SYSTEM FOR LICENSING DIGITAL WORKS

(76) Inventors: **James C. Hillegass**, Wayzata, MN (US); **Yaobing Deng**, Flossmoor, IL (US); **Mark Eastom**, New Brighton, MN (US); **Richard R. Fritz**, Maple Grove, MN (US); **John C. Gateley**, Plymouth, MN (US); **James A. Grinsfelder**, St. Paul, MN (US); **Stephen A. Grove**, Minneapolis, MN (US); **Eric Steven Hockett**, Minneapolis, MN (US); **Nikolay G. Sokratov**, Bellevue, WA (US); **James G. Swanson**, Eagan, MN (US); **John S. Thompson**, Afton, MN (US); **Boris Mamedov**, Plymouth, MN (US); **James A. Nordgaard**, Minneapolis, MN (US); **Paul E. Onnen**, Sammamish, WA (US)

Correspondence Address:
BECK AND TYSVER P.L.L.C.
2900 THOMAS AVENUE SOUTH
SUITE 100
MINNEAPOLIS, MN 55416 (US)

(21) Appl. No.: **11/323,209**

(22) Filed: **Dec. 30, 2005**

Related U.S. Application Data

(63) Continuation-in-part of application No. 09/844,475, filed on Apr. 27, 2001, now Pat. No. 7,076,468.

(60) Provisional application No. 60/200,230, filed on Apr. 28, 2000. Provisional application No. 60/200,193, filed on Apr. 28, 2000.

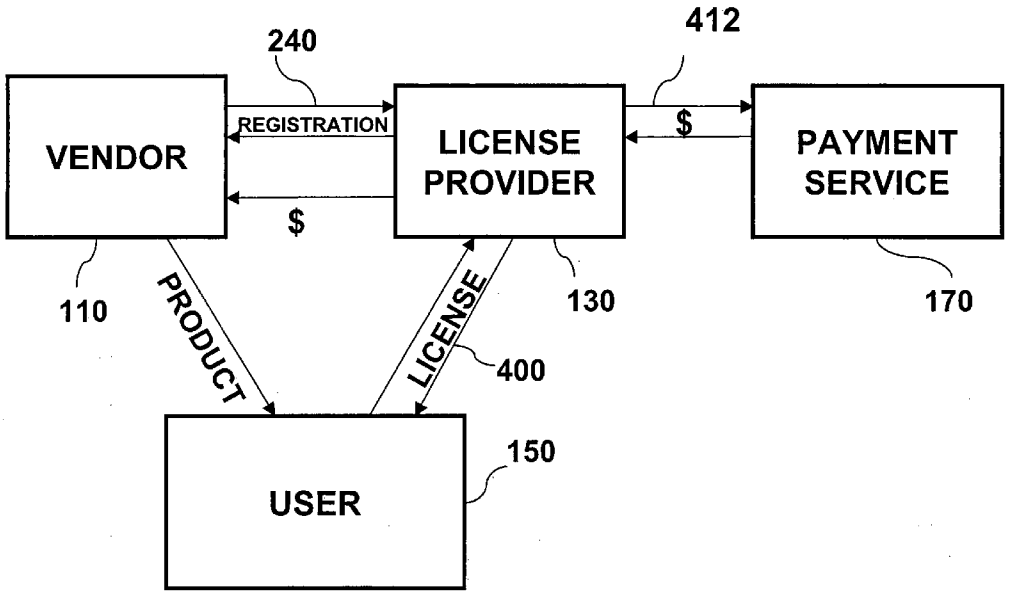
Publication Classification

(51) **Int. Cl.**
G06Q 99/00 (2006.01)

(52) **U.S. Cl.** **705/59**

(57) **ABSTRACT**

A method and system is presented for a digital licensing scheme that separates the license from the digital file containing the copyrightable material. According to the present invention, the files can be downloaded from any server, and transferred from user to user, even after the file has been licensed. The present invention utilizes producer software running on a vendor's computer, server software running on a computer provided by the license provider, and player software operating on the user's computer. Digitally encrypted communication streams keep communications between the producer software, the license provider, and the player software confidential. A software component running on the user's computer checks to make sure that the appropriate product license has been purchased. This is accomplished by comparing the product ID in the product license with the product ID contained in the product file. The software also checks that the person seeking to play the product file is the user that actually paid for the license. This is accomplished by comparing the user ID in the product license with a user ID in a user license. Finally, an operating system ID found in the user license is compared with the same information obtained from the currently running operating system, to ensure that the user license was created for the currently operating computer.



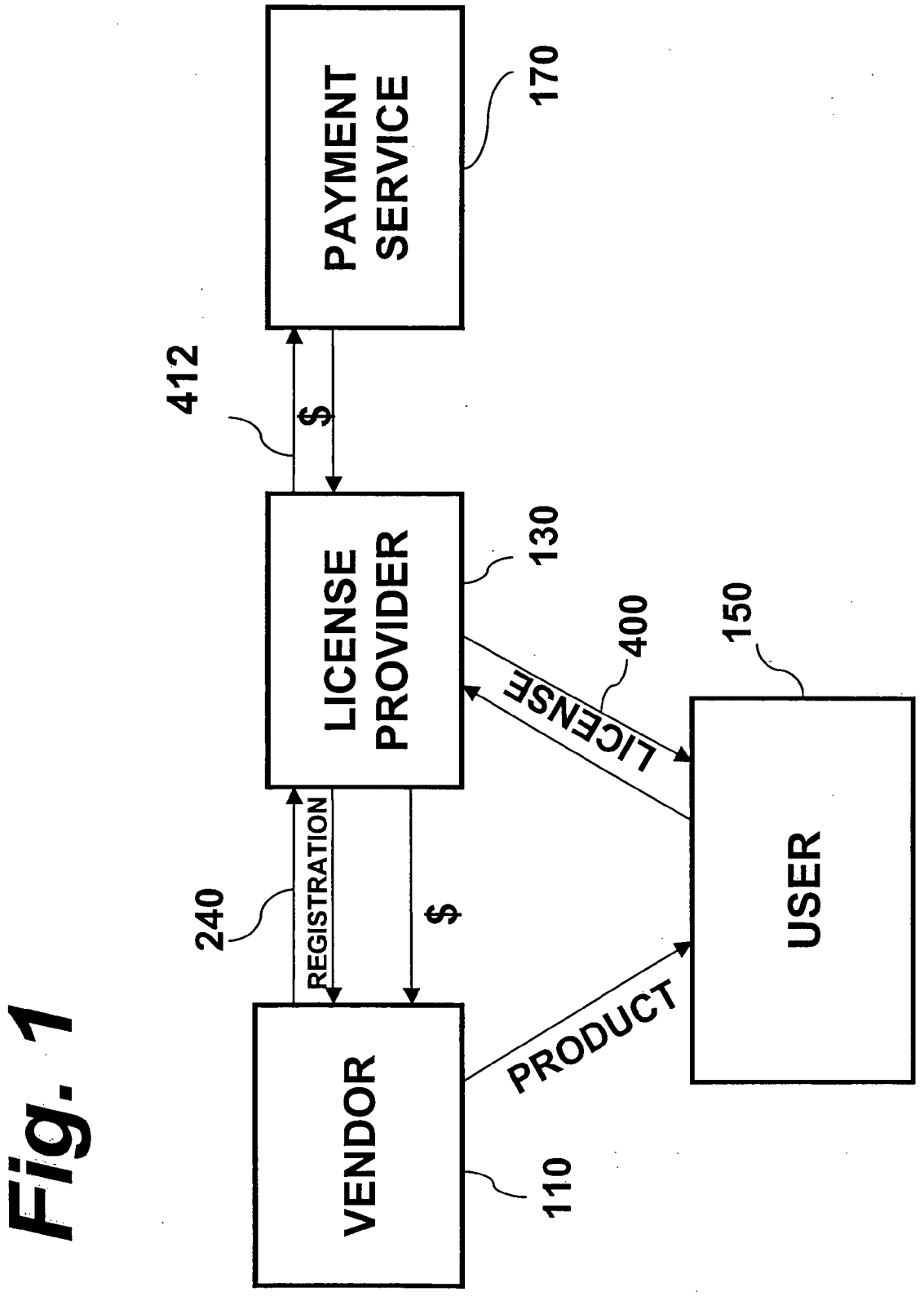
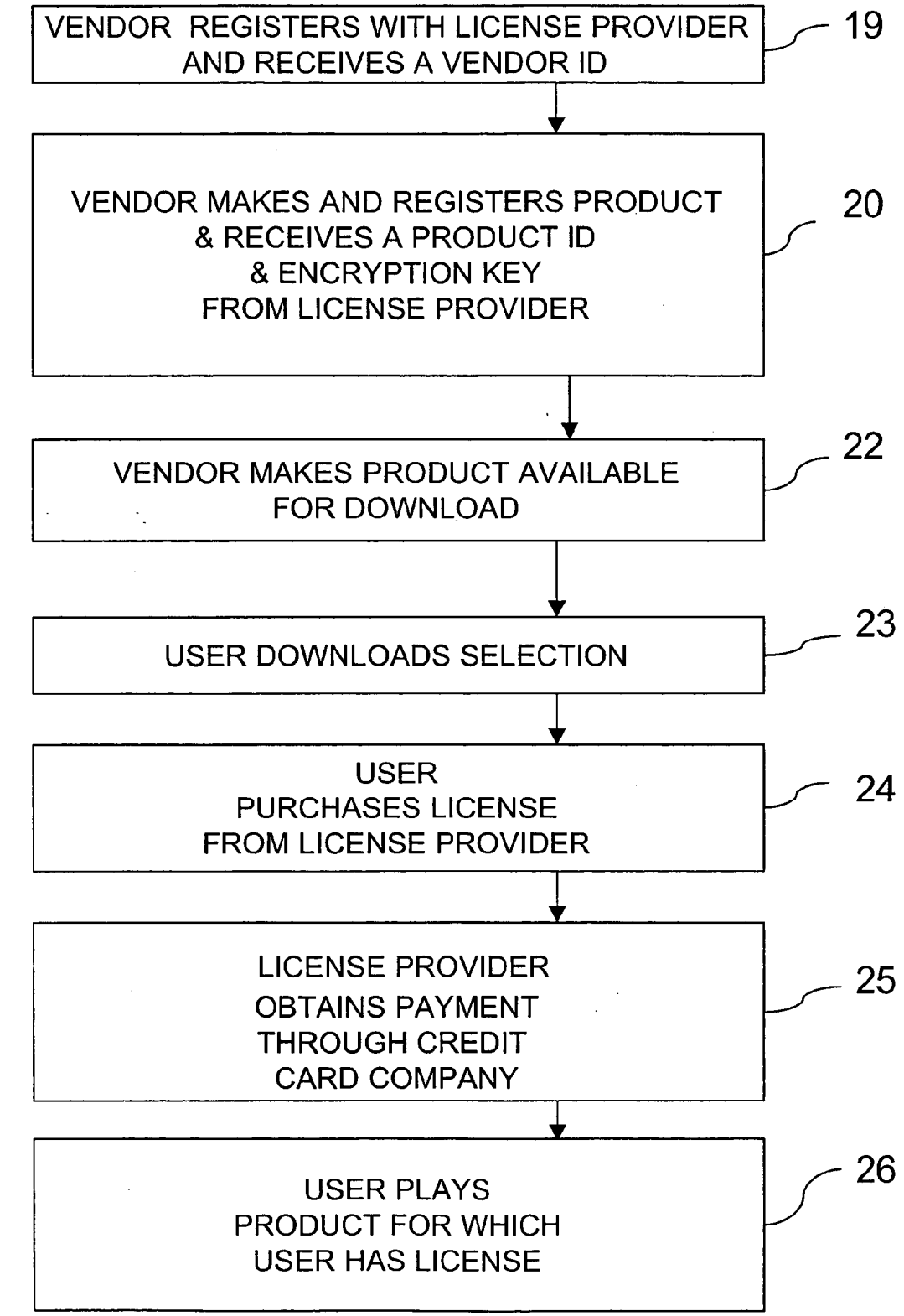

PAYMENT SERVICE
170
412
$
LICENSE PROVIDER
130
400
LICENSE
150
USER
REGISTRATION
240
$
PRODUCT
VENDOR
110

Fig. 1

Fig. 2


VENDOR REGISTERS WITH LICENSE PROVIDER AND RECEIVES A VENDOR ID
19
VENDOR MAKES AND REGISTERS PRODUCT & RECEIVES A PRODUCT ID & ENCRYPTION KEY FROM LICENSE PROVIDER
20
VENDOR MAKES PRODUCT AVAILABLE FOR DOWNLOAD
22
USER DOWNLOADS SELECTION
23
USER PURCHASES LICENSE FROM LICENSE PROVIDER
24
LICENSE PROVIDER OBTAINS PAYMENT THROUGH CREDIT CARD COMPANY
25
USER PLAYS PRODUCT FOR WHICH USER HAS LICENSE
26

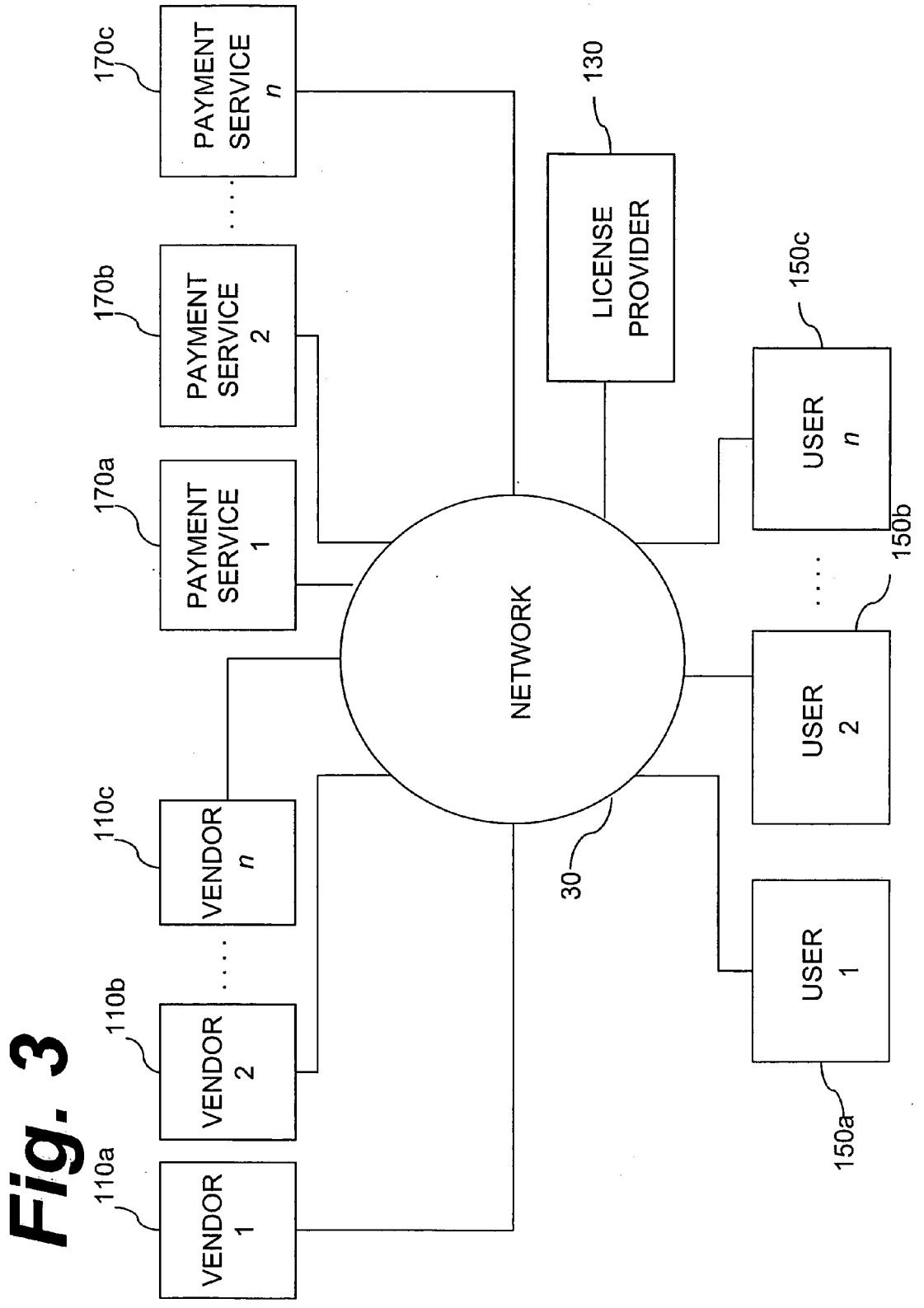

VENDOR 1
110a
VENDOR 2
110b
VENDOR n
110c
PAYMENT SERVICE 1
170a
PAYMENT SERVICE 2
170b
PAYMENT SERVICE n
170c
NETWORK
30
LICENSE PROVIDER
130
USER 1
150a
USER 2
150b
USER n
150c

Fig. 3

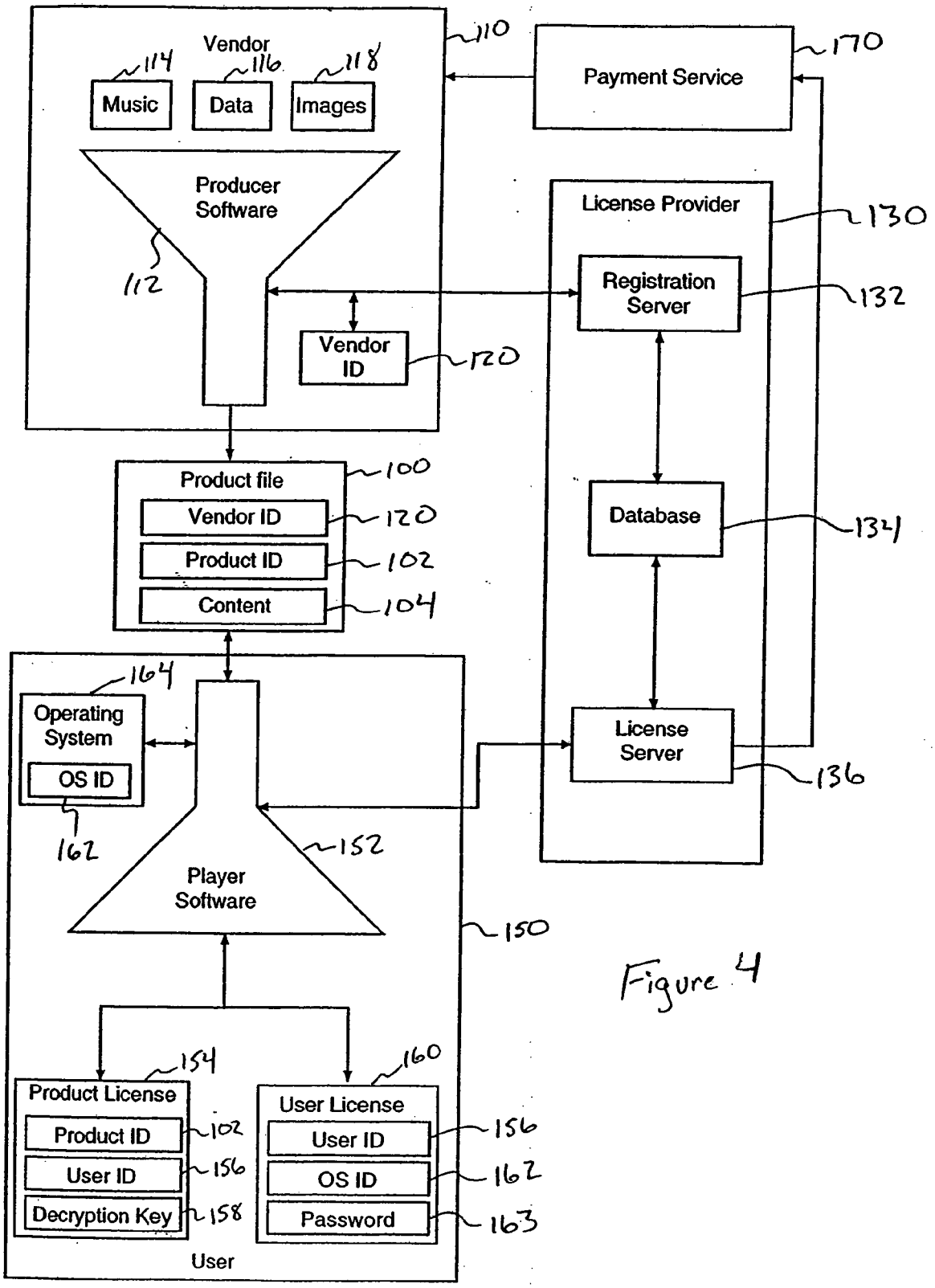

Vendor
110
Music
114
Data
116
Images
118
Producer Software
112
Vendor ID
120
Payment Service
170
License Provider
130
Registration Server
132
Database
134
License Server
136
Product file
100
Vendor ID
120
Product ID
102
Content
104
Operating System
164
OS ID
162
Player Software
152
150
Product License
154
Product ID
102
User ID
156
Decryption Key
158
User License
160
User ID
156
OS ID
162
Password
163
User

Figure 4

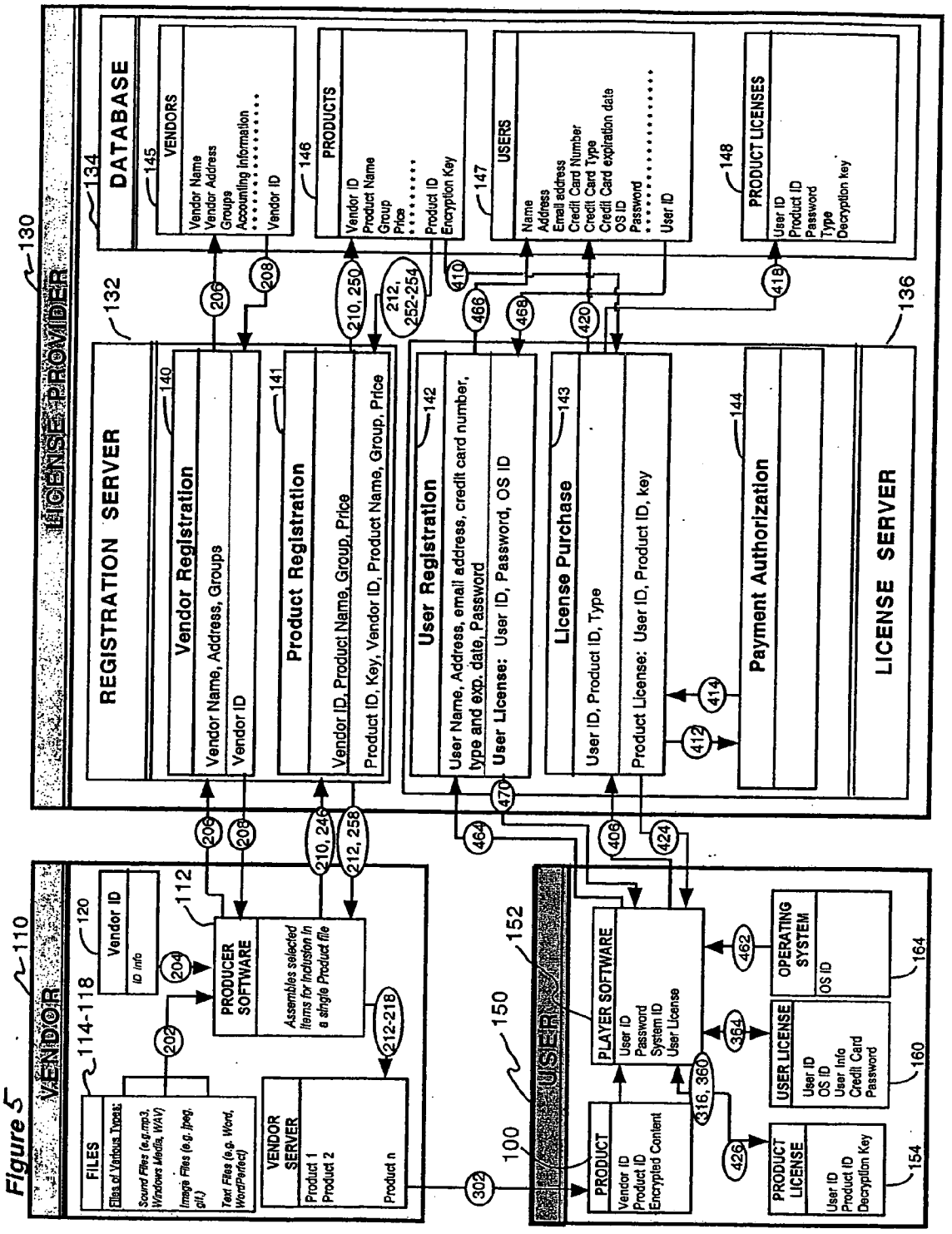


Figure 5
110
VENDOR
114-118
FILES
Files of Various Types:
Sound Files (e.g.mp3, Windows Media, WAV)
Image Files (e.g. jpeg, gif.)
Text Files (e.g. Word, WordPerfect)
120
Vendor ID
ID info
112
PRODUCER SOFTWARE
Assembles selected items for induction in a single Product file
VENDOR SERVER
Product 1
Product 2
Product n
212-218
202
204
206
208
210, 246
212, 258
302
100
150
152
USER
PRODUCT
Vendor ID
Product ID
Encrypted Content
PLAYER SOFTWARE
User ID
Password
System ID
User License
316, 360
364
462
426
OPERATING SYSTEM
OS ID
164
USER LICENSE
User ID
OS ID
User Info
Credit Card
Password
160
PRODUCT LICENSE
User ID
Product ID
Decryption Key
154
464
406
424
470
412
414
LICENSE PROVIDER
130
REGISTRATION SERVER
132
Vendor Registration
140
Vendor Name, Address, Groups
Vendor ID
Product Registration
141
Vendor ID, Product Name, Group, Price
Product ID, Key, Vendor ID, Product Name, Group, Price
User Registration
142
User Name, Address, email address, credit card number, type and exp. date, Password
User License: User ID, Password, OS ID
License Purchase
143
User ID, Product ID, Type
Product License: User ID, Product ID, key
Payment Authorization
144
LICENSE SERVER
136
DATABASE
134
VENDORS
145
Vendor Name
Vendor Address
Groups
Accounting Information
Vendor ID
PRODUCTS
146
Vendor ID
Product Name
Group
Price
Product ID
Encryption Key
USERS
147
Name
Address
Email address
Credit Card Number
Credit Card Type
Credit Card expiration date
OS ID
Password
User ID
PRODUCT LICENSES
148
User ID
Product ID
Password
Type
Decryption key
210, 250
212, 252-254
410
466
468
420
418

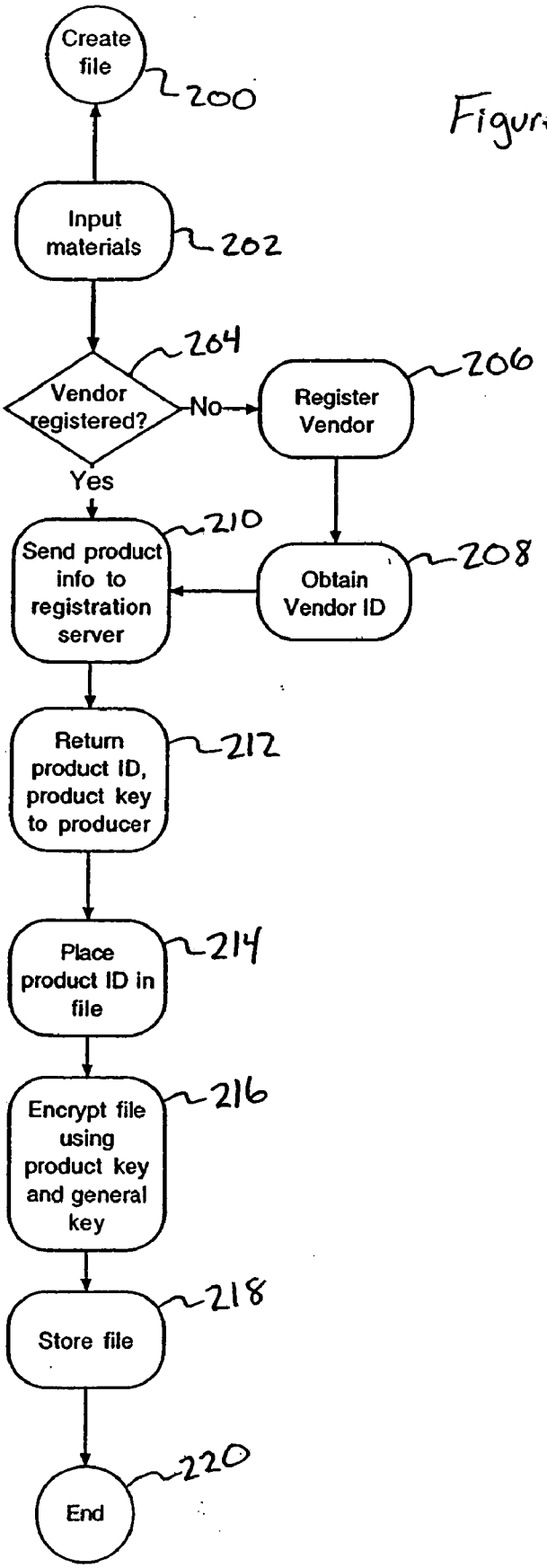

Create file
200
Input materials
202
Vendor registered?
204
No
Register Vendor
206
Yes
Send product info to registration server
210
Obtain Vendor ID
208
Return product ID, product key to producer
212
Place product ID in file
214
Encrypt file using product key and general key
216
Store file
218
End
220

Figure 6

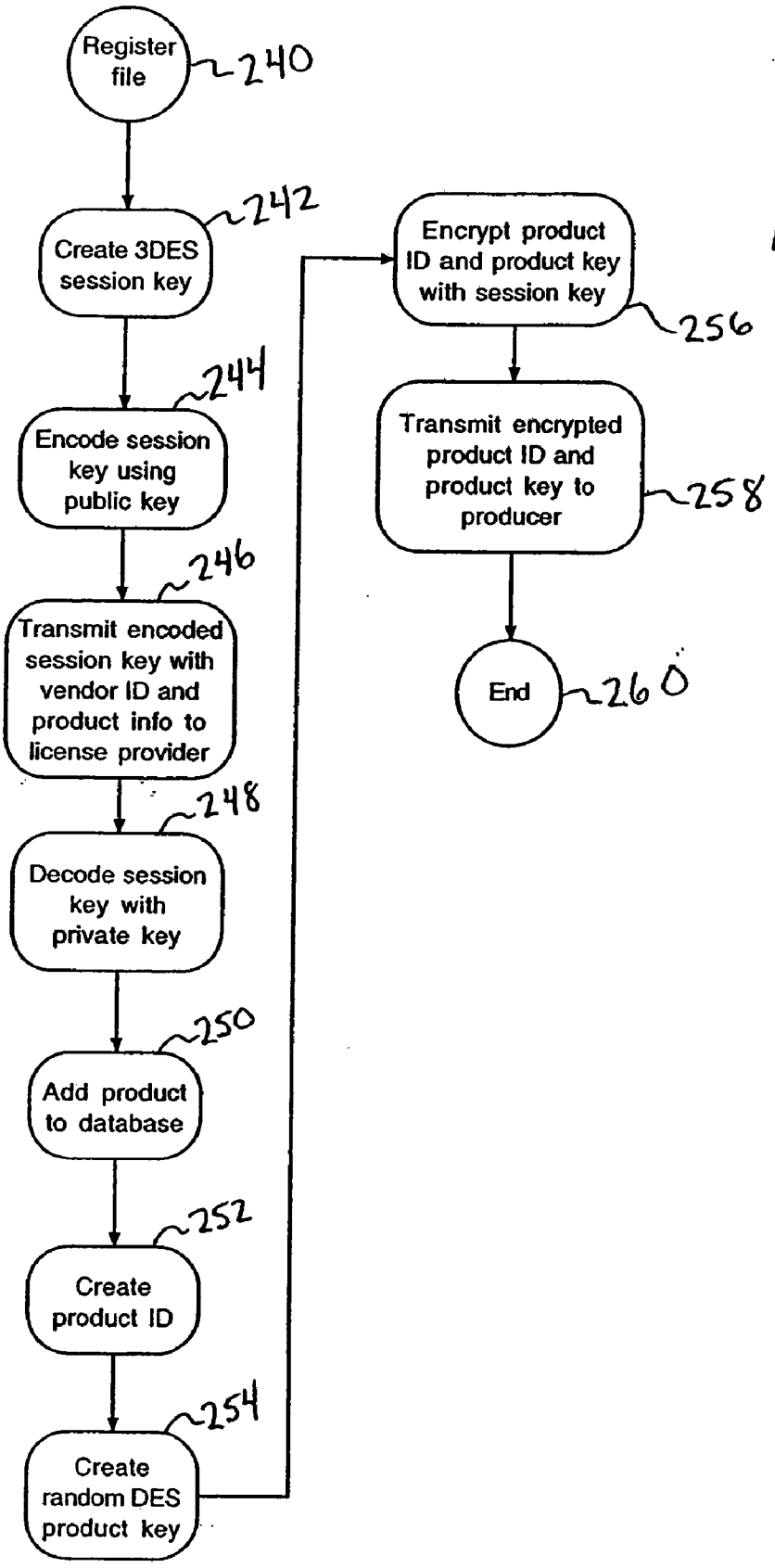

Register file
240
Create 3DES session key
242
Encode session key using public key
244
Transmit encoded session key with vendor ID and product info to license provider
246
Decode session key with private key
248
Add product to database
250
Create product ID
252
Create random DES product key
254
Encrypt product ID and product key with session key
256
Transmit encrypted product ID and product key to producer
258
End
260

Figure 7

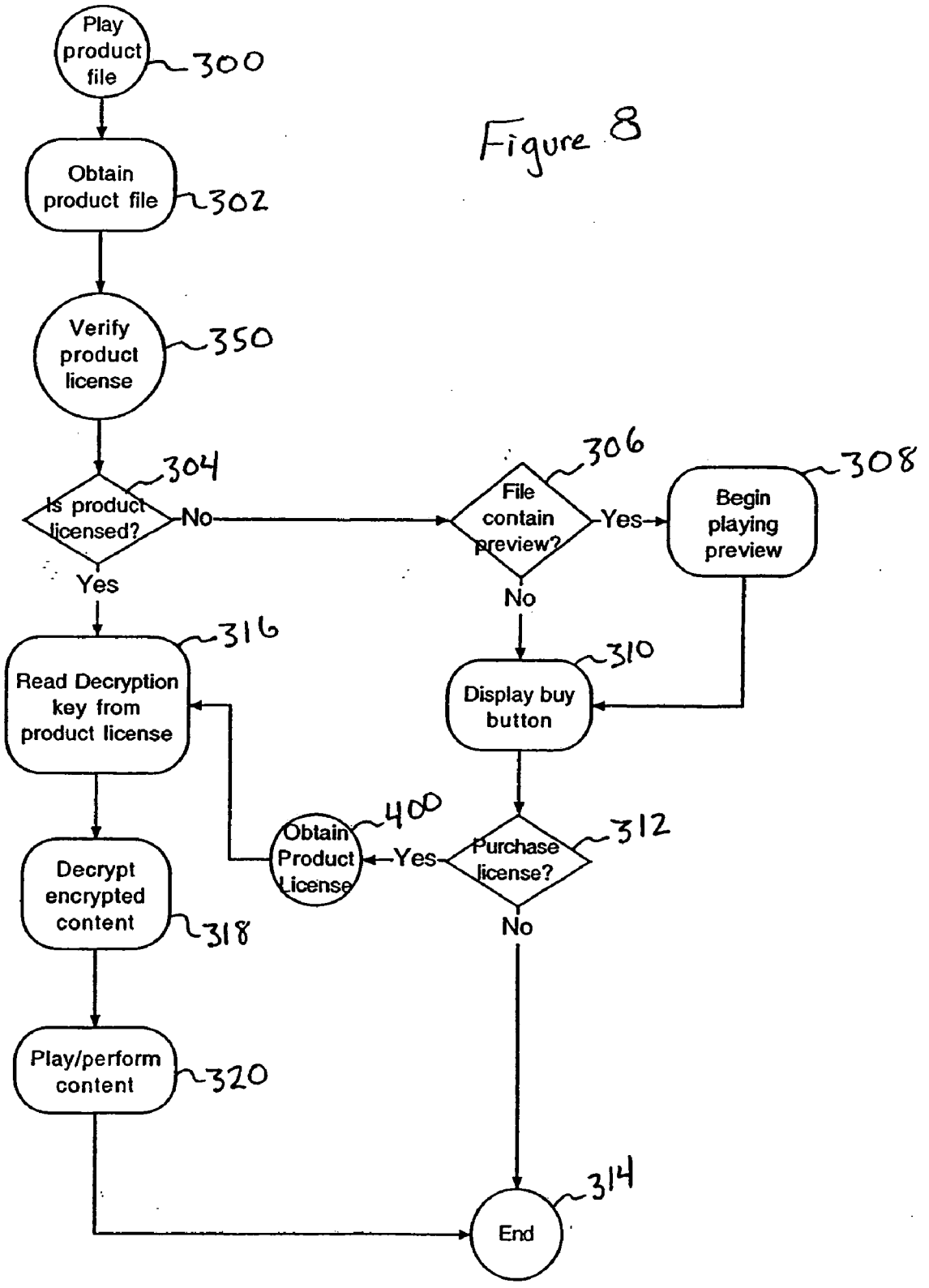

Play product file
300
Obtain product file
302
Verify product license
350
Is product licensed?
304
No
Yes
File contain preview?
306
Yes
Begin playing preview
308
No
Read Decryption key from product license
316
Display buy button
310
Obtain Product License
400
Yes
Purchase license?
312
Decrypt encrypted content
318
No
Play/perform content
320
End
314

Figure 8

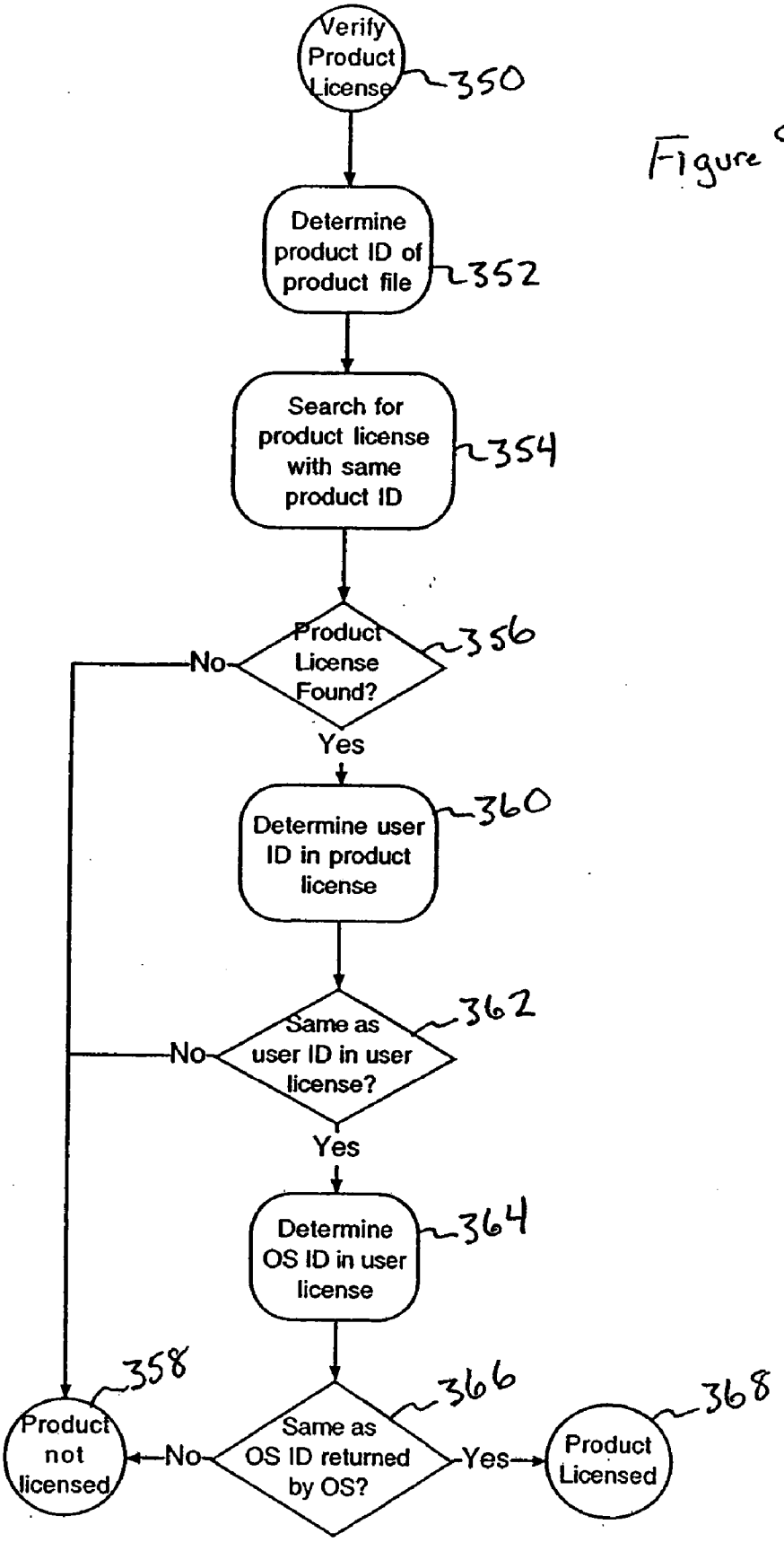

Verify Product License
350
Determine product ID of product file
352
Search for product license with same product ID
354
Product License Found?
356
No
Yes
Determine user ID in product license
360
Same as user ID in user license?
362
No
Yes
Determine OS ID in user license
364
Same as OS ID returned by OS?
366
No
Yes
Product not licensed
358
Product Licensed
368

Figure 9

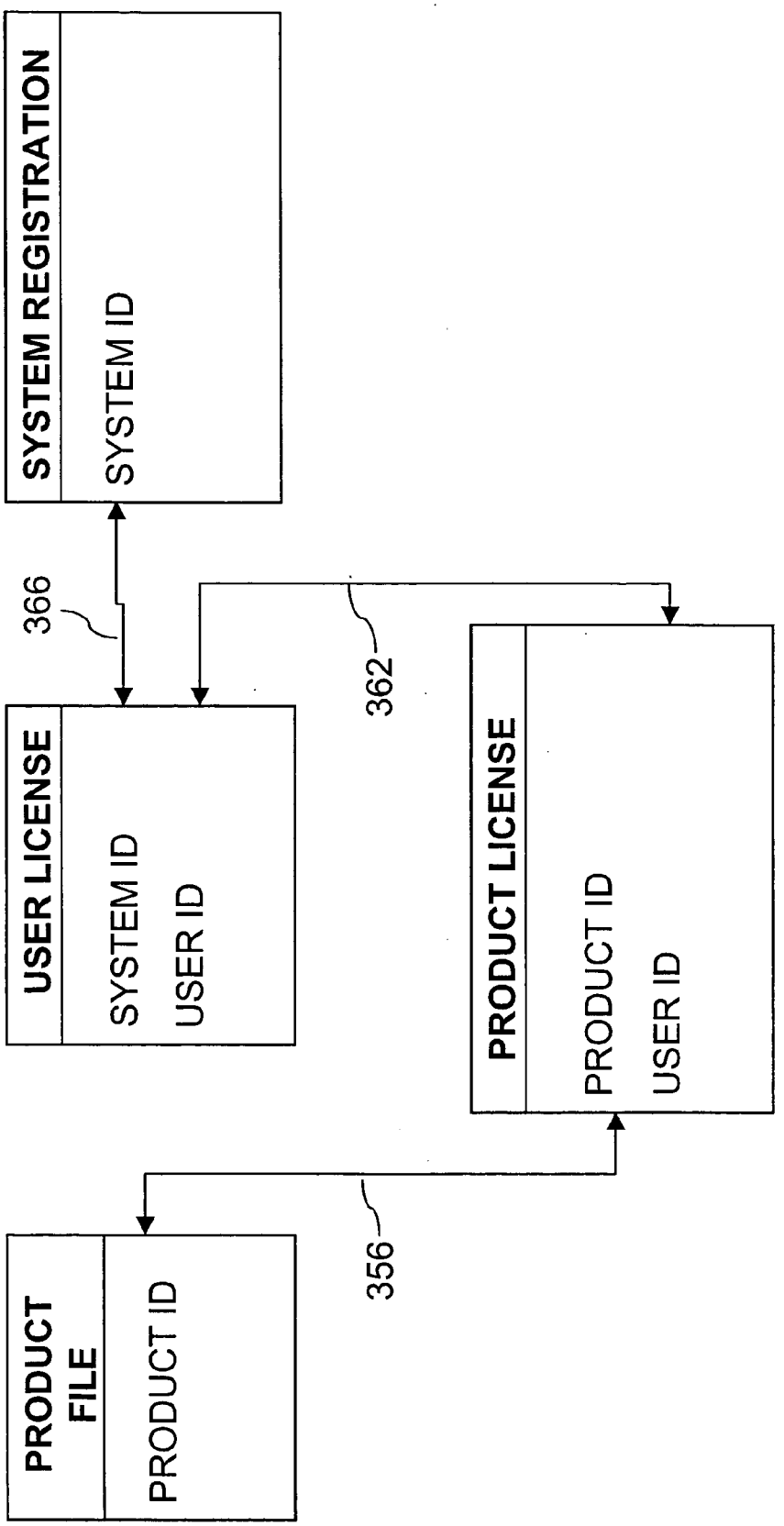
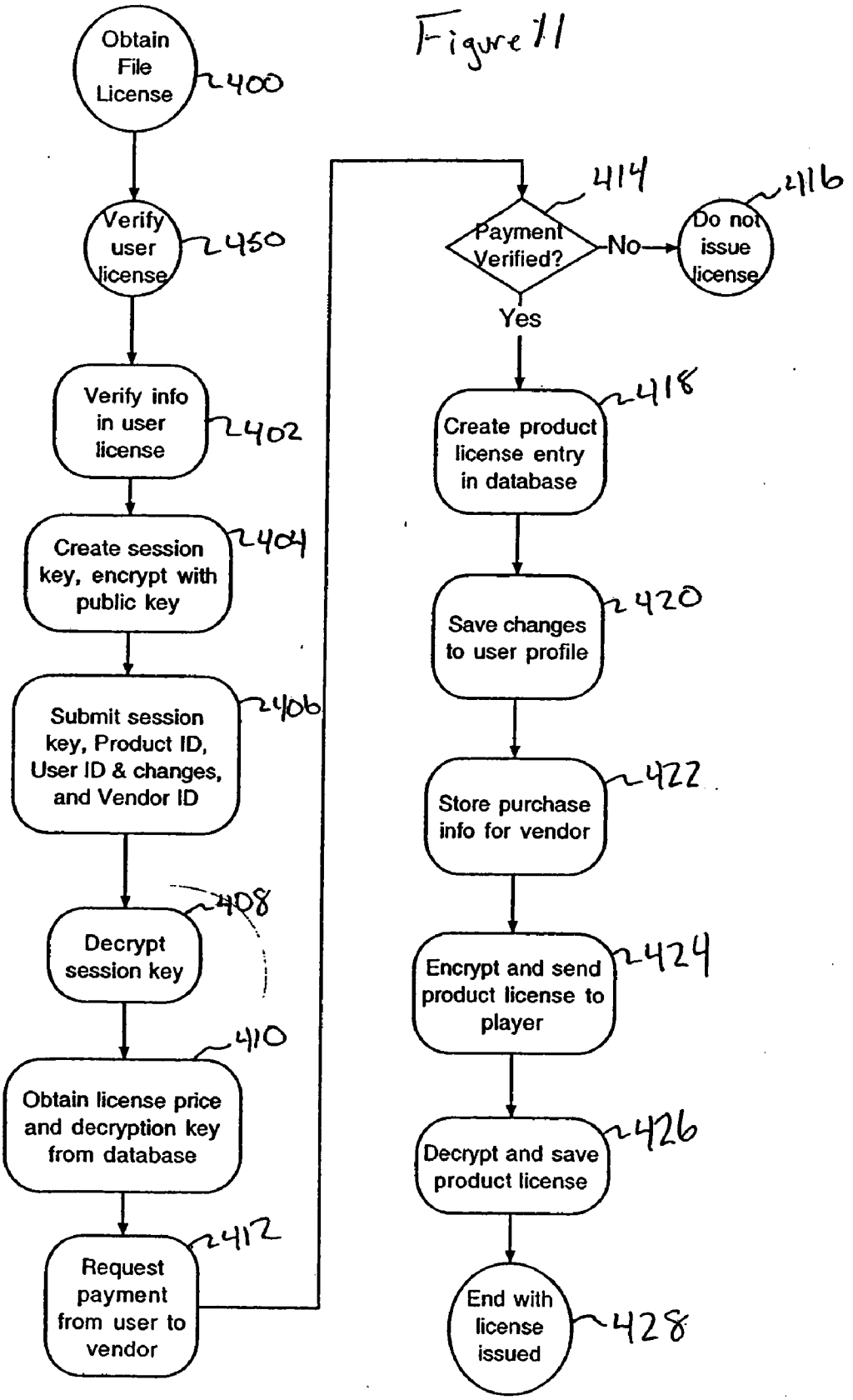

SYSTEM REGISTRATION
SYSTEM ID
366
USER LICENSE
SYSTEM ID
USER ID
362
PRODUCT LICENSE
PRODUCT ID
USER ID
356
PRODUCT FILE
PRODUCT ID

Fig. 10

Figure 11


Obtain File License
400
Verify user license
450
Verify info in user license
402
Create session key, encrypt with public key
404
Submit session key, Product ID, User ID & changes, and Vendor ID
406
Decrypt session key
408
Obtain license price and decryption key from database
410
Request payment from user to vendor
412
Payment Verified?
414
No
Do not issue license
416
Yes
Create product license entry in database
418
Save changes to user profile
420
Store purchase info for vendor
422
Encrypt and send product license to player
424
Decrypt and save product license
426
End with license issued
428

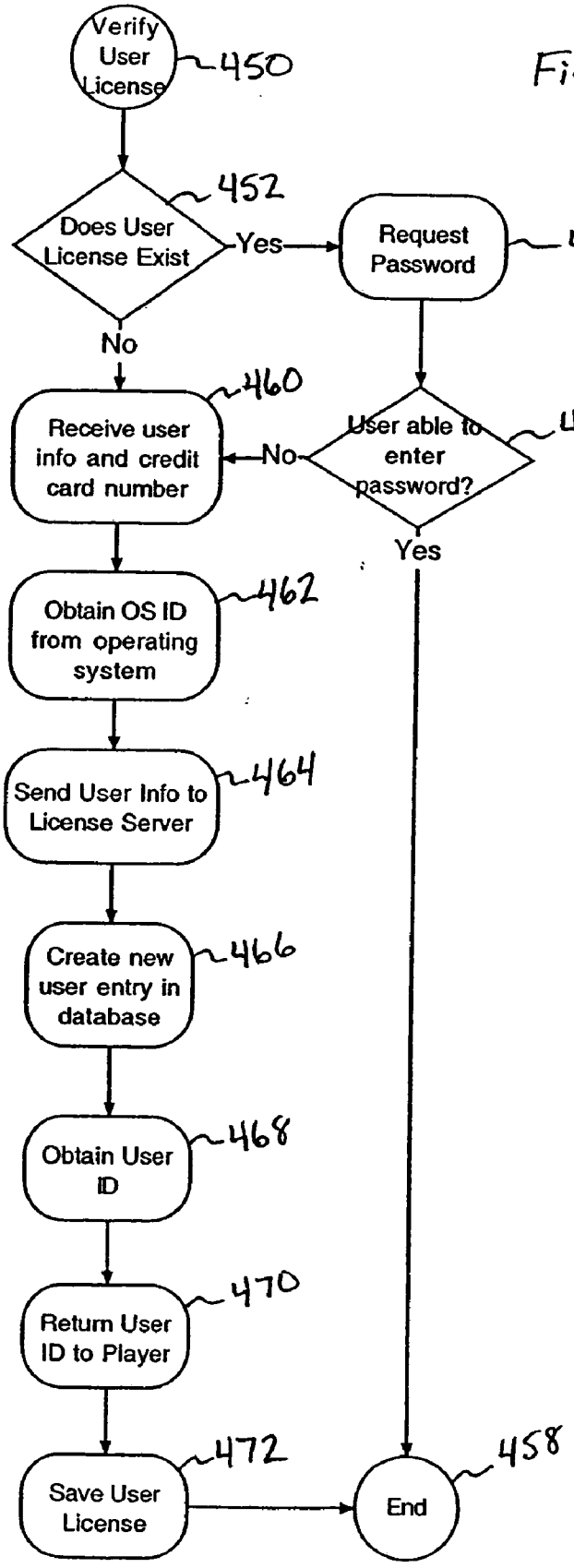

Verify User License
450
Does User License Exist
452
Yes
Request Password
454
No
User able to enter password?
456
No
Yes
Receive user info and credit card number
460
Obtain OS ID from operating system
462
Send User Info to License Server
464
Create new user entry in database
466
Obtain User ID
468
Return User ID to Player
470
Save User License
472
End
458

Figure 12

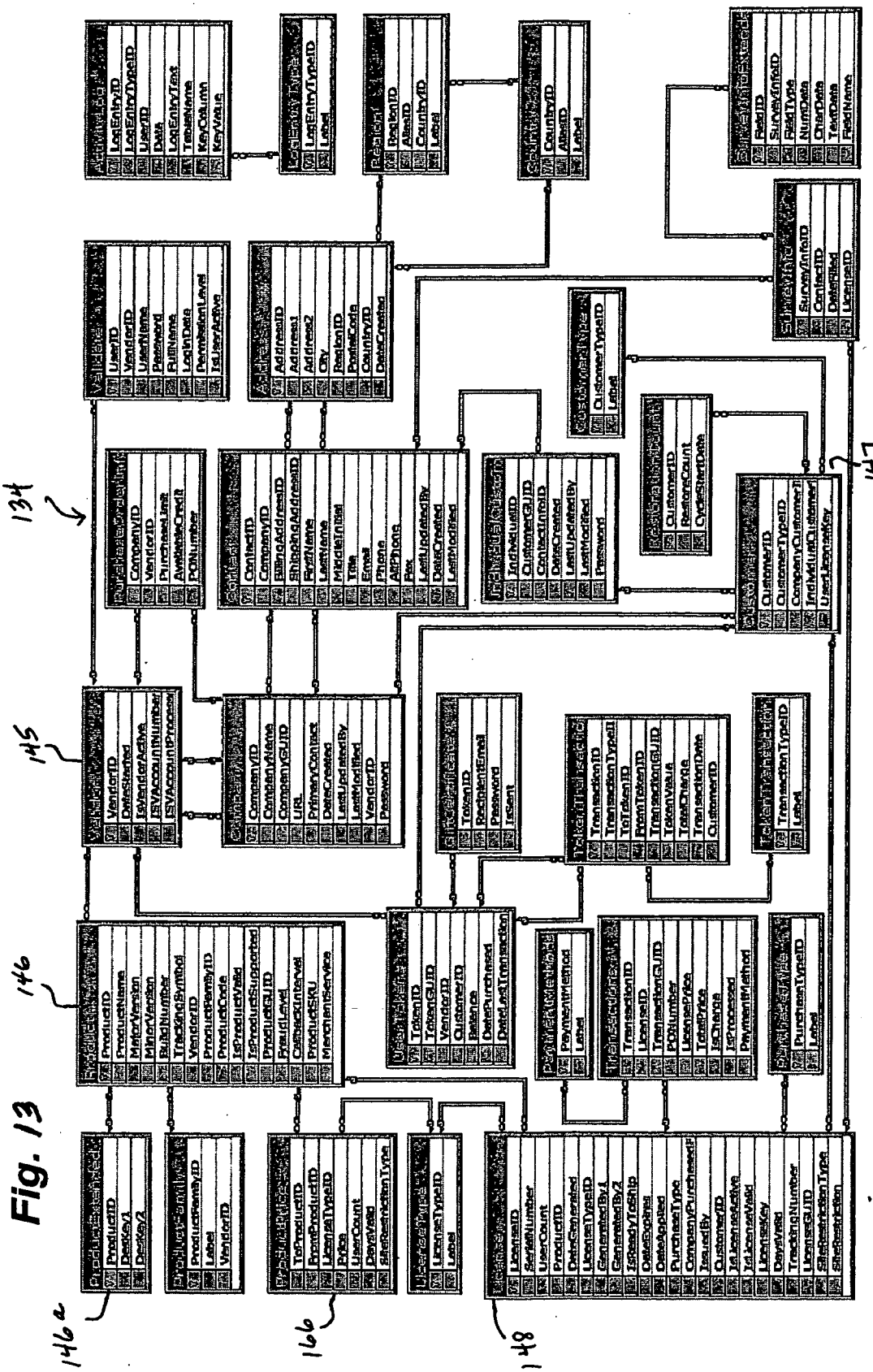


Fig. 13

METHOD AND SYSTEM FOR LICENSING DIGITAL WORKS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Ser. No. 09/845,041, filed Apr. 27, 2001 and to U.S. Ser. No. 09/844,475, filed Apr. 27, 2001 and claims the benefit of U.S. Provisional Application Ser. No. 60/200,230, filed on Apr. 28, 2000, and U.S. Provisional Application Ser. No. 60/200,193, filed on Apr. 28, 2000.

FIELD OF THE INVENTION

[0002] The present invention relates generally to a system and method for controlling access to copyrighted materials in a digital format. More particularly, the present invention relates to a system for creating and maintaining licenses that exist separate from the copyrighted materials.

BACKGROUND OF THE INVENTION

[0003] The widespread demand for music and the growing availability of the Internet as a means of commerce have resulted in a multi-billion dollar industry for digital content including music via the Internet. Audio compression technologies such as MP3 (MPEG Layer III) have allowed digital music to be stored at compression rates of 10-1 or better. This compression technology, along with the rise of the Internet and increasing bandwidth, have led to an explosion of downloadable digital music available over the Internet. Individual tracks of music can now be downloaded from the World Wide Web, sent via e-mail, or stored and downloaded via FTP sites and Usenet newsgroups.

[0004] This new technology has brought new challenges to the policing of copyright interests in materials distributed in or convertible to digital form. Unauthorized copying of digital materials is of particular concern in the music industry, though efforts have been made to prevent it. One approach is to control access to the digital files, requiring the receipt of payment before the file can be downloaded. To prevent redistribution of files that have been downloaded, technology has been applied in attempt to limit the ability to access the files to a particular computer.

[0005] U.S. Pat. No. 5,765,152 to Erickson ("Erickson '152") describes a system and method for managing copyrighted electronic media. Erickson '152 describes the use of a registration system to make documents available over a computer network, and an authorization system for end-users to obtain the desired level of permission to use and alter the document. End users are then able to subsequently register the resulting derivative work. According to the Erickson '152 system, permissions are attached to the document file, and the user downloads or accesses the document file with the appropriate permissions attached to the document file. Thus, the permissions must co-exist with the documents. This is disadvantageous for a number of reasons. For example, if the user loses a document file, he/she also loses their permission to use the document. Further, Erickson's system contemplates distribution of documents through specific servers, i.e. the author does not have the option of posting the document from any server he/she chooses and this may be insufficient to meet the author's marketing objectives. Finally, once the document is downloaded and licensed, it cannot be further distributed since the site-specific license is embedded in the file.

SUMMARY OF THE INVENTION

[0006] What is needed is a secure, digital licensing scheme that allows easy and widespread distribution of copyrightable materials, while at the same time preventing subsequent unauthorized access. Further, it would be advantageous for an authorized user to transport licensed materials between several computers. Finally, what is needed is a secure and convenient method of distributing music files, where a producer of the music can distribute files to potential customers without having to attend to licensing and selling functions.

[0007] The present invention provides a digital licensing scheme that separates the license from the digital file containing the copyrightable material. According to the present invention, the files can be downloaded from any server, and transferred from user to user, even after the file has been licensed.

[0008] The present invention utilizes producer software running on a vendor's computer, server software running on a computer provided by the license provider, and player software operating on the user's computer. Digitally encrypted communication streams keep certain communications between the producer software, the license provider, and the player software confidential.

[0009] A software component running on the user's computer checks to make sure that the appropriate product license has been purchased before allowing access to a digital product file. This is accomplished by comparing the product ID in the product license with the product ID contained in the product file. The software also checks that the user seeking to play the product file is the user that actually paid for the license. This is accomplished by comparing the user ID in the product license with a user ID in a user license. Finally, an operating system ID found in the user license is compared with the same information obtained from the currently running operating system, to ensure that the user license was created for the currently operating computer.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] **FIG. 1** is a schematic illustration of the system and method of the present invention;

[0011] **FIG. 2** is a flow chart describing the process for creating a Product, distributing and licensing the Product, and using a licensed Product;

[0012] **FIG. 3** is a schematic illustration showing how multiple Vendors and Users are coordinated through the system and method of the present invention;

[0013] **FIG. 4** is a schematic illustration of the major components of the present invention.

[0014] **FIG. 5** is a schematic illustration of the present invention showing the flow of data through the components.

[0015] **FIG. 6** is a flow chart showing the process for creating a file.

[0016] **FIG. 7** is a flow chart showing the process for registering a file.

[0017] **FIG. 8** is a flow chart showing the process for playing a product file.

[0018] **FIG. 9** is a flow chart showing the process for verifying a product license.

[0019] **FIG. 10** is a schematic illustration of the security checks made to verify that a Product License authorizes the playing of a given Product, according to the system and method of the present invention.

[0020] **FIG. 11** is a flow chart showing the process for obtaining a product license.

[0021] **FIG. 12** is a flow chart showing the process for verifying a user license.

[0022] **FIG. 13** is a schematic illustration of the tables comprising the database used in the preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

1. Overview

[0023] As illustrated in rudimentary form in **FIG. 1**, the system and method of the present invention coordinates the activities of an author, artist or producer ("Vendor") **110**, an end user ("User") **150** of the copyrighted materials, a "License Provider" ("License Provider" or "LP") **130**, and an entity for processing payment transactions including credit card purchases, debit card withdrawals, electronic cash distribution or the like ("Payment Service") **170**. The basic steps in a method according to this invention are illustrated in **FIGS. 1 and 2**. The Vendor **110** registers itself with the License Provider (step **19**). The Vendor **110** creates a product and registers the Product **10** with the License provider **130** (**20**). (Step **20** is described below in considerably greater detail as processes **200** and **240**.) The Vendor **110** makes the Product available to Users **6** on or through Electronic Media, such as via the Internet, ftp, CD, or e-mail (step **22**). The User **150** downloads selected Products from the Vendor **110** and is able to view a preview of the contents of the Product (**23**). If the User **150** wants to view and own the right to use the entire contents, the User **150** then purchases a license from the License provider **130** for that Product (**24**). (Step **24** is described in greater detail below as process **400**.) The License provider **130** in turn passes the User's credit card information through a Credit Card Processor **8** or other transaction agent to obtain payment (**25**). (Step **25** is described in greater detail below as process **412**.) The License Provider then sends, or makes available for download, a License for that User for the Product. The User **150** is able to fully play and view the Product **10** (**26**) subject to constraints determined by the type of license purchased. The License provider **130** pays the Vendor **110** for sales of its registered Products (**27**).

[0024] As illustrated in **FIG. 3**, the system and method of the present invention accommodate multiple Vendors **5***a*-**5***c*, multiple Users **6***a*-**6***c*, and multiple Credit Card Processors **8***a*-*c*. In a preferred embodiment, the Vendors **5** store Products **10** on servers and make Products **10** available to Users **6** over a network **30**, such as the Internet, for download onto their personal computer hard drives or other mechanism for storing electronic or digital data. The License provider **130** stores license and Product information, but not necessarily the Products **10** themselves, on a server. The License provider **130** makes licenses available for Users **6** to purchase over the Internet. The License provider **130** is networked, either through a dedicated connection or through the Internet to Credit Card Processors **8**.

[0025] As shown in **FIG. 4**, the present invention provides a method and system for creating, playing, and licensing digital content files **100**. For the purpose of example, the present invention will be described in the context of files containing digital music tracks. However, the present invention is equally applicable to files containing any type of digital material for which licensing is desired.

[0026] In the preferred embodiment, there are four parties who utilize aspects of the present invention. The first is the vendor **110**, who supplies the source materials and creates the music file **100**. The second party is the remote license provider **130**, who is responsible for providing information for the creation and licensing of file **100**. The third party is the user **150**, who receives the file **100** from the vendor **110** and licenses the file **100** from the remote license provider **130**. Finally, a payment service **170** ensures payment of a license fee to the vendor **110** when the license provider **130** has provided a license to the user **150**. The communication between these entities could occur through any standard communication protocol. In the preferred embodiment, communication between remote computing applications is accomplished through remote procedure calls, or RPCs. Note that the functions performed by each of these entities would be fundamentally the same even if one entity took on the functions of one or two other entities shown in **FIG. 4**. The present invention would not be altered by such a combination of functions in single entity.

[0027] The vendor **110** could be a music producer, a record label, an independent band, or any other party who has the right to duplicate and distribute the content placed in file **100**. The vendor **110** creates the file **100** using a producer program **112**, which is represented in **FIG. 4** with a funnel. This representation illustrates that the producer **112** takes numerous and disparate sources of content and combines them into a single file **100**.

[0028] As illustrated in **FIG. 4**, producer **112** can accept as input multiple tracks of music **114**, data **116**, and images **118**. The data **116** included in the file **100** could include lyrics, liner notes, UPC Codes for a CD, or information about the music such as the name of the musician(s), the title of the music collection and its individual tracks, etc. The images **118** may be still images that the vendor **110** wishes to have displayed whenever the file **100** is played. Additionally, the images **118** may be photographs of musicians, video images, cover art, or any other type of multi-media content.

[0029] The format of the inputted materials **114**-**118** is immaterial to the present invention, as the materials **114**-**118** can either be converted by the producer program **112** to a preferred format in the product file **100**, or the materials **114**-**118** can simply be stored in the file **100** in their original format. For instance, music data **114** can be provided in any known music format such as traditional CD audio format or a standard waveform format such as WAV, AIFF, or AU. The producer software **112** would preferably save the music data **114** in a compressed format such as MP3. Images can be stored in any of the well-known compressed file types such

as JPEG or GIF. Video images **118** can also be added and stored in a compressed format such as AVI (Video for Windows), MPEG, or Quicktime.

[**0030**] The producer program **112** is in communication with the license provider **130**, specifically the registration server **132** operated by the license provider **130**. The vendor **110** is identified to the license provider **130** by including its unique vendor ID **120** in its communications. The registration server **132** can be physically located on the same or nearby computer used by vendor **110** for the producer software **112**. Ideally, however, the registration server **132** is remotely located, and in communication with multiple producer programs **112**. The license provider **130** also operates a database **134**, which stores information about vendors **110**, users **150**, product files **100**, and licenses; and a license server **136**, which is used to control the licensing of product files **100**.

[**0031**] While the registration server **132**, database **134**, and license server **136** are illustrated as separate entities in **FIG. 4**, it is well within the scope of the invention to combine these services into one or two separate entities. For instance, a single application running on a single computer could provide all of the functionality of the registration server **132**, database **134**, and license server **136**. Alternatively, the two servers **132**, **136** could be combined and communicate with database **134**. It is even within the scope of the present invention to have multiple registration servers **132** and license servers **136** functioning simultaneously.

[**0032**] More detail about the registration server **132**, database **134**, and the license server **136** can be seen in **FIG. 5**. Registration server **132** has two main components, vendor registration **140**, and product registration **141**. License server **136** has three main components; namely a user registration component **142**, a license purchase component **143**, and a payment authorization component **144**. These components are simply one way of dividing the functions of the two servers **132**, **136**. Many ways are possible and within the scope of the present invention.

[**0033**] Similarly, in the preferred embodiment, database **134** contains entries (tables or sub-databases) for at least the following types of data: vendors **145**, products **146**, users **147**, and product licenses **148**. More detail concerning the tables in the database **134** of the preferred embodiment can be seen in **FIG. 13**, as described below.

[**0034**] Returning to **FIG. 4**, the product file **100** that is created by the producer software **112** contains the music **114**, data **116**, and images **118** that were entered into producer **112**. This content **104** is stored in an encrypted format in the file **100**. File **100** also contains a vendor ID **120** that indicates the vendor **110** who created the file **100**, as well as a product ID **102** that uniquely identifies the product file **100**. The vendor **110** can make the product file **100** available to users **150** in a variety of manners well known in the prior art, such as through download from a web site or via FTP. The vendor **110**, the license provider **130**, or any other party can host these sites, since there is no need for the party hosting the product file **100** to be a license provider **130**. The product file **100** is not altered after creation by the vendor **110**. Consequently, the product file **100** can be freely transferred from user **150** to user **150**, with each user being able to separately license the file **100**.

[**0035**] The user **150** can access the content **104** on the product file **100** using player software **152**. In the preferred embodiment, where the content **104** of file **100** contains music **114** and related materials **116**, **118**, the player software **152** is capable of playing the music **114** to end users, while also allowing users access to the lyrics, images, and other content **104** in file **100**. A sophisticated player **152** would also be able to take a UPC code from the product file **100** and electronically search various audio/video Internet-based retailers for the availability and price of physical copies (such as a CD) of the music collection in file **100**.

[**0036**] To have total access to the encrypted content **104** in file **100**, the user **150** will have to obtain a product license **154** that contains the decryption key **158** specific for that product file **100**. The product license is obtained by interaction between the player software **152** and the license server **136**. Alternatively, product licensing could be handled at the user **150** level by a different program operating on the same computer as, and in conjunction with, the player software **152**. For ease in description, the player software **152** will be described as having both playback capabilities and license handling capabilities, although it would be well within the scope of the present invention to split these actions into two separate interacting programs.

[**0037**] Because a product license **154** is also specific for a particular user **150**, the user **150** must obtain their own user license **160** from license server **136** before any products **100** can be licensed. The product license **154** and the user license **160** are both stored at the computer of the user **150** as well as in the database **134** of the license provider **130**. In order to protect the product license **154** and user license **160** from unauthorized access and alteration, both licenses **154**, **160** are protected with triple-DES ("3DES") encryption. The product license **154** is limited to a specific product file **100** because the product license contains the product ID **102**. The product license **154** is also limited to a particular user by containing a user ID **156**, which is also found in the user license **160**. The user license **160** is limited to a particular user **150** in part by tying the user license **160** to identifying information **162** stored in the operating system **164** of the user's computer. Because in the preferred embodiment the user license **160** will contain credit card numbers and other confidential information of the user **150**, the user license **160** will be protected by password **163**.

[**0038**] As part of the license process, the user **150** will authorize that a payment be made in return for the license. Thus, before the user license **160** is returned to the user, the license server **136** will contact the payment service **170** to collect payment from the user **150**. Typically, this will be done through either a credit card transaction or through some type of electronic cash or some similar Internet payment system. The payment service **170** is generally capable of directly crediting an account belonging to the vendor **110** that created the product file **100**.

2. File Creation Process **200**

[**0039**] **FIG. 5** shows the flow of data through the various components of the system. This **FIG. 5** is best viewed in light of the flow charts found in **FIGS. 6 through 12**. Where possible, the steps found in the flow charts are shown with arrows on **FIG. 5**, with the step reference numeral on or near the arrow.

[**0040**] The procedure for creating file **100** is shown as process **200** in **FIG. 6**. First, the vendor **110** accumulates in

the producer program **112** the materials **114-118** that will be combined into file **100**, as seen in step **202**. The producer program **112** will then contact the registration server **132**, and the registration server **132** determines whether the vendor **110** needs to register as a new vendor (step **204**). Alternatively, producer program **112** could merely search for vendor ID **120** on its local computer to determine if it needs to register. If the vendor **110** has not previously registered, vendor **110** provides information about itself, which is used by the registration server **132** to create a vendor entry in database **134** (step **206**). In this process, registration server **132** assigns a vendor ID **120** to the vendor **110** (step **208**). The vendor ID **120** is then stored both in the database **134** in the vendor record **145** and in the computer used by vendor **110**. The vendor ID **120** is preferably stored in the operating system registry of the computer used by the vendor **110**. It is also preferred to allow the vendor **110** to freely move the vendor ID **120** to multiple computers, thereby allowing the vendor **110** to make music files **100** from multiple locations or through multiple employees. The vendor ID **120** is then used in all later communications between the registration server **132** and the vendor **110**.

[**0041**] Alternatively, rather than requiring information from the vendor **110** at the time the vendor entry is made into the database **134**, an entry can be made with no information merely to create a vendor ID **120**. The vendor **110** could then be allowed to enter and edit information about itself and its product files **100** at a later date, such as by logging in with the vendor ID **120** at a web site.

[**0042**] Once the vendor **110** is registered as a vendor, the producer software **112** contacts the registration server **132** and sends to server **132** its vendor ID **120** and information about the file **100** being created (step **210**). The information sent will include the product name, the license fee amount, and the category or group in which the vendor **110** wishes to locate the file product **100**. These categories can be universal categories, or, preferably, be categories created and separately maintained for each vendor **110**.

[**0043**] The registration server **132** then creates a product entry **146** in the database **134** and returns the information need by the producer software **112** to create file **100** (step **212**). Specifically, the registration server **132** returns a product ID **102** and a DES encryption key. The details surrounding the submission of product information to the registration server and the return of the product ID **102** and encryption key (steps **210** and **212**) are described in more detail below in connection with **FIG. 7**.

[**0044**] The producer program **112** inserts the received product ID **102** in the file **100** being created (step **214**). To ensure against unauthorized access to the music in file **100**, at least the music information is encrypted with the product specific DES encryption key received from the registration server **132** (step **216**). In the preferred embodiment, encryption is also used to protect header sections of file **100**. The encryption of header sections is preferably done with a general DES encryption key that is the same with all copies of producer program **112**, rather than the product specific DES key returned by the registration server **132**. The header section contains basic information about the file, including title and musician, and also the checksums that guarantee the integrity of the content **104**. The preferred embodiment also uses headers to define basic information about each track of music contained in file **100**. These track headers are also compressed with the DES key known to all producer applications **112** as well as all player software **152**.

[**0045**] After the encryption is finished in step **216**, the producer software **112** saves the complete music file **100** (step **218**). The process of creating file **100** is then complete, as shown as step **220**.

3. File Registration Process **240**

[**0046**] The details of the file registration process **240** are set forth in **FIG. 7**. The first step **242** is for the producer software **112** to create a unique 3DES encryption key for the upcoming communication session with the license provider **130**. The 3DES encryption algorithm is a symmetrical encryption system. Thus, this newly created 3DES session key must be communicated to the license provider **130** before 3DES encryption can be used for communication. In order to transmit this session key to the license provider **130** in a secure fashion, the session key is itself encrypted with a public encryption key whose matching private key is known only to the license provider **130** (step **244**).

[**0047**] The encrypted session key is then transmitted along with the product information and the vendor ID **120** to license provider **130**, as shown in step **246**. The license provider **130** then uses its private key to decrypt the 3DES session key created by producer software **112** (step **248**).

[**0048**] The next step **250** is to create a new product entry **146** into database **134** using the information transmitted along with the session key in step **246**. When a new product is entered into database **134**, the license provider **130** creates a product ID **102** and stores this ID **102** with the other product information in database **134** (step **252**). In addition to the product ID **102**, the license provider **130** also generates a random DES encryption key that will serve as the product encryption key (step **254**). This product encryption key is also stored with the product information in database **134**.

[**0049**] It is now necessary to transmit the newly generated product ID **102** and product encryption key back to the producer software **112**. In order to transmit this information securely, it is encrypted using the session key that was previously generated by producer software **112** (step **256**). Once this is accomplished, the encrypted product ID **102** and the product encryption key can be transmitted back to vendor **110** (step **258**), and the register file process is completed (step **260**).

[**0050**] 4. Playing a Product File **300FIG. 8** shows the process **300** for playing a product file **100**. The process **300** starts by the user **150** obtaining the product file **100** created by vendor **110** (step **302**). Typically, this is done by downloading the file **100** from a web site sponsored by vendor **110**, license provider **130**, or any other source. In addition, since the file **100** is not changed during the license process, user **150** can obtain the file **100** from any other user **150**, regardless of whether the other user **150** had licensed the product **100** or not.

[**0051**] The next step in playing the file **100** is for the player software **152** to determine whether or not user **150** has a valid product license **154** for the file **100**. This is done in process **350**, which is described below in more detail in connection with **FIG. 9**. Player **152** takes different steps

depending on whether a valid product license exists, which is analyzed in step **304**. If there is no valid product license **154**, the product file **100** is examined to determine whether any preview content exists in the file (step **306**). If there is preview content, that content is then played by the player **152** in step **308**.

[**0052**] While the preview is playing, the player **152** should then present user **150** with the option to purchase a product license **154** for the file **100**. This is done in step **310**, which is also performed even if the file **100** did not contain preview information. If the user **150** does not wish to license the file **100** (as determined at step **312**), then the process **300** for playing a file **100** is completed (step **314**). If the user **150** does choose to purchase a product license **154** for the product **100**, then process **400** for obtaining a file license is performed. Process **400** is described below in more detail in connection with **FIG. 11**.

[**0053**] Whether a valid product license **154** is determined to exist at step **304**, or whether a new product license **154** is purchased through process **400**, it is possible to then play the complete contents **104** of the product file **100**. This is accomplished by reading the decryption key from the product license **154** in step **316**, and then decrypting content **104** with this key in step **318**. The decrypted content **104** is then performed by player **152** in step **320**, and the process **300** completes at step **314**.

5. Verifying an Existing Product License **350**

[**0054**] The process **350** of verifying an existing product license is shown in the flowchart of **FIG. 9**. The first step **352** is to examine the product file **100** to determine the product ID **102**. The player **152** then examines all of the product licenses **154** available to the user **150** in search for a product license **154** that contains the same product ID **102** (step **354**). The product licenses **154** can be stored on the computer of user **150** in a variety of ways. For instance, each product license **154** could exist in its own independent file. Alternatively, the product license **154** could form part of a registry or other service database maintained by the operating system **164** of the computer. The product licenses **154** could even consist of an entry in a database, plain file, or structured file that is maintained by player software **152** in a customized format.

[**0055**] After searching, it must be determined if any applicable product licenses **154** were found (step **356**). If not, the process **350** has determined that the product **100** is not licensed, and the process **350** ends with that result in step **358**. If a product license **154** was found containing the correct product ID **102**, then that product license **154** is examined to determine the user ID **156** for that license **154** (step **360**). The user license **160** for the current user **150** is then examined to see if its user ID **156** matches the user ID **156** of the product license **154** (step **362**). If not, the product **100** is not properly licensed and the process **350** ends at step **358**.

[**0056**] If the user IDs **156** match, the player software **152** then examines the operating system ID **162** that was stored with the user license **160** (step **364**). This OS ID **162** is then compared to the identification that is returned live from the operating system **164**. The OS ID **162** is basically some identification that is unique to the currently operating computer or the current user of the operating computer. For example, in the Windows 95/98 operating system from Microsoft Corporation (Redmond, Wash.), the OS ID **162** can be the registered user's name for the operating system. While different operating systems have different types of system values that are retrieved in different ways, the player software **152** should be able to extract some type of identifying information from the operating system **164** in which it operates. If step **366** determines that the two retrieved OS IDs **162** do not match, then the process **350** ends with no valid license at step **358**. If the OS IDs **162** do match, process **350** ends by returning a value indicating that a valid license for the file **100** has been found (step **368**).

[**0057**] This last step of examining the OS IDs **162** is useful in verifying that the user license **160** was created or otherwise appropriate for this computing environment. This helps to prevent the "sharing" of user licenses **160** between differing users **150**. However, since the user license **160** will contain personal, private financial information about a user **150**, namely the user's credit card information, there is already a strong disincentive against sharing a user license **160**. Thus, it would be well within the scope of the present invention to skip steps **364** and **366** in process **350**, and rely on the existence of private information in the user license **160** to prevent the sharing of user licenses **160**.

[**0058**] **FIG. 10** schematically depicts the steps of comparing product IDs **356**, user IDs **362**, and OS or system IDs **366**.

6. Obtaining a File License **400**

[**0059**] The process **400** for obtaining a product file license **154** is shown in the flowchart of **FIG. 11**. Before anything else in process **400**, the player software **152** must verify that the current user **150** is known to the license server **136**. This is done by checking for and verifying the current user license **160**, a process **450** which is described in detail below in connection with **FIG. 12**.

[**0060**] Once a valid user license **160** has been identified by process **450**, the information in the user license **160** will be presented to the user **150** for verification (step **402**). Of course, this step **402** could optionally be skipped if the user **150** had just created their user license **160** in process **450**. Generally, the information will be presented visually to the user **150** in this step **402**, and the user **150** will be given the opportunity to change any of the relevant information. Among the information shown will be the credit card number that was previously used by the user **150**. Because most users **150** would be very reluctant to let others see their credit card number, the showing of the number to the user **150** at this stage should serve as a deterrent to users **150** sharing their user licenses **160** and their passwords with other users. In addition to a credit card number, it is well within the scope of the present invention to use other private information for payment purposes and for providing a disincentive toward sharing a user license. Examples of such information include a bank account number, gift certificate number, a debit card number, and a stored value card number. Non-financial related information could also be used solely to help prevent the sharing a user license, including a social security number, or even a home address and telephone number.

[**0061**] Once the user **150** has validated the information from their user license **160**, the player software **152** ran-

domly generates a new 3DES session key. This session key will be used to encrypt the information contained in the product license **154** that will be retrieved from the license server **136**. Because the 3DES encryption scheme is a symmetrical encryption scheme, and because the player software **152** randomly generates the 3DES key, it is necessary to securely transmit this new key to the license server **136**. This is accomplished by encrypting this new key using a public key for which only the license server **136** knows the matching private key. This is all accomplished in step **404**.

[**0062**] The player software **152** next submits to the license server **136** a request for a new product license **154** (step **406**). This submission includes the appropriate product ID **102**, the user ID **156** of the user **150**, the vendor ID **120** found in file **100**, the encrypted 3DES session key, and any changes to the user profile made by user **150**.

[**0063**] The license server **136** will then decrypt the session key with its private key (step **408**). The next step **410** is to access the product information stored in database **134** to obtain the license price and decryption key for the product file **100**. Although the license price is probably also stored with product file **100**, it may be wise to verify this license price against the database even if the license price was submitted along with other information in step **406**. The vendor ID **120** can also be verified against the vendor ID **120** associated with the product entry in database **134**. Alternatively, the vendor ID **120** could be excluded from the submission of step **406**, with the vendor ID **120** simply being determined through the database **134**. Of course, the decryption key (which is the same as the encryption key created in step **254**) is stored only in database **134** and is not found in product file **100**.

[**0064**] In step **412**, the license server **136** then requests that the payment service **170** make a payment from the user **150** in favor of the vendor **110** identified by the vendor ID **120**. In the preferred embodiment, all communications by the license server **136** to the payment service **170** are handled by the payment authorization component **144**, as shown in **FIG. 5**. Typically, the payment authorization component **144** uses external credit card gateways as the payment service **170**. The license server **136** can submit the payment request as if the request is coming from any of the vendors **110** that might be identified in the vendor ID **120**. In this way, payment will be made directly from the payment service **170** to the vendor **110**. Typically, the license provider **130** will collect some payment for its service. When the payment from the payment service **170** goes directly to the vendor **110**, the license provider **170** must track these license purchases in its database and the regularly bill the vendor **110**. Alternatively, the payment request can be made in favor of the license provider **130** itself. In this case, the license provider **130** will track license purchases in its database and make regular payments to vendors **110**.

[**0065**] The payment authorization component **144** can do some validity preprocessing of the payment information before submission of the request to the payment service **170**. Examples of preprocessing that are done in the preferred embodiment of the present invention include verifying the structure of the credit card number, such as by examining the starting digit and the total number of digits.

[**0066**] The payment service **170** will then indicate to the license server **136** whether payment was actually made. If step **414** indicates that no payment was made (for instance, because the credit card number was invalid), the process for obtaining a file license **400** terminates at step **416** with no license issued.

[**0067**] If the payment if verified, then the license server **136** creates a product license entry **148** into database **134** (step **418**). At a minimum, the license entry will contain the product ID **102**, the user ID **156** and the decryption key **158**. It is possible to develop a license that is has limitations in it, such as date limitations or site limitations. If such limitations are desired, those limitations would be inserted into the database **134** as part of the license entry **148**. The limitations would also appear inside the product license **154**. It would be up to the player software **152** to interpret and enforce license limitations when it reads a product license **154** containing such limitations.

[**0068**] The license server **136** should also save to database **134** any changes to the user data that were submitted in step **406**. This is done in step **420**. In addition, it may be useful to maintain data on all licenses furnished by the license server **136** for purposes of both billing the vendor **110** and to allow vendor to see product license information and trends. Information that would allow this kind of tracking, such as customers' names, dates of purchase and total purchase amounts, is stored in a transactions database entry made to database **134** in step **422**.

[**0069**] The license server **136** must then return the product license **154** to the player software **152** (step **424**). In order to ensure secure transit of the product license **154**, the product license **154** is first encrypted using the 3DES session key generated in step **404**. When the product license **154** is received by player software **152**, it is decrypted with the session key and then saved for later use in step **426**. The product license **154** is always stored in an encrypted format to keep it protected. The process of obtaining a file license **400** is then completed with the license issued at step **428**.

7. Verifying a User License **450**

[**0070**] The process for verifying a user license is shown as process **450** in **FIG. 12**. The first step **452** is to determine whether a user license **160** exists. If so, the user **150** is asked to enter the password **163** for the user license **160** (step **454**). If the user is successfully able to enter the password **163** that was stored with the user license **160**, which is checked in step **456**, then the user license **160** has been verified and process **450** terminates at step **458**.

[**0071**] If a user license **160** does not exist, or if the user **150** is not able to successfully enter a password, then it is necessary to create a new user license **160**. This is done by having the user **150** enter personal information such as name, address, e-mail address, as well as a password **163** and a valid credit card number (step **460**). The player software **152** will then obtain the OS ID **162** from the operating system **164** (step **462**). All of this information is then transmitted to the license server **136** in step **464**.

[**0072**] Upon receipt of a request for a new user license **160**, the license server **136** will create a new entry in the users portion **147** of database **134** (step **466**). When this is done, the license server **136** or the database **134** generates a new user ID **156** and saves it in the database with the user information (step **468**). The newly created user ID **156** is then transmitted back to the player software **152** along with

the other components of the user license **160**, including the OS ID **162** and the password **163** (step **470**). Alternatively, only the user ID **468** could be returned and then combined with the information obtained by the player software **152** in steps **460** and **462** to create the user license **160**. The last step **472** is to save the user license **160** so that it can be retrieved at a later date. The user license **160** will be stored in an encrypted format, preferably using the 3DES technology. The process **450** then terminates at step **458**.

8. License Restoration Process

[**0073**] Users **150** are authorized to transfer user licenses between machines a limited number of times. If the license is transferred without any interaction with the player software **152** or the license server **136**, the transfer will be unsuccessful because a user license **160** is tied to a specific machine through the OS ID **162**. If the license were merely moved without changing the embedded OS ID **162**, there would not be a match in step **366**, and the user license **160** would be ineffectual.

[**0074**] To accomplish the transfer of user licenses **160**, the player software **152** has the ability to save the license information to a safe location such as a floppy disk. If the hard disk containing the user license **160** then crashes, the user **150** can restore the user license **160** through the player software **152**. To do so, the player software **152** requires the user **150** to enter the correct password **162**. Then the player software **152** contacts the license server with request to recover a user license **152**. This request would contain basically the same information sent to the license server **136** in step **464**, including the new OS ID **162**, as well as the User ID **156** that is being recovered. Assuming that user has not restored their user license **160** more than the pre-determined limit, the license server **136** will return a new user license **160** that will work with the new OS ID **162**.

[**0075**] The license server **136** keeps track of the number of times license restoration is attempted by a user **150**. A limit is placed on how many times one can restore licenses from the license server **136**. If credit card numbers are not always required to obtain a user license **160**, then a lower limit for restorations can be placed on users **150** whose user license **160** does not contain credit cart information. Using this technique, it is possible to move a user license **160** to a different computer, albeit only limited number of times.

[**0076**] If a hard drive is lost, not only is the user license **160** lost, but so also are all of the product license **154** that were on the drive. Consequently, player software **152** also allows a user **150** with a valid user license to query database **134** and download all known product licenses **154** for the user's user ID **160** that are not currently on the hard drive. In this way, a user can secure his or her licenses merely by backing up the user license to a floppy disk through the utility provided by player software **152**. It is also possible in this manner to have a duplicate set of user license **160** and product licenses **154** on multiple computers.

9. Database **134**

[**0077**] As shown in **FIG. 5**, database **134** contains numerous sub-databases or tables, including vendors **145**, products **146**, users **147**, and product licenses **148**. A more complete definition of the database **134** is shown in **FIG. 13**. As seen in that figure, the database **134** is a relational database comprising many related tables, such as vendor table **145**, product table **146**, user table (labeled "Customer") **147**, and product license table (labeled "License") **148**. Because of the relational nature of the preferred embodiment of database **134**, some of the information shown in a single table in **FIG. 5** is actually contained in multiple tables in **FIG. 13**. For instance, the decryption keys are actually stored in a "Product Extended" table **146***a*, while the product price is actually stored in a related "Product Price" table **146***b*.

[**0078**] Although an illustrative version of the system and method is shown, it should be clear that many modifications to the system and method may be made without departing from the scope of the invention. For instance, the flow charts described above requested that a user **150** enter the password stored in the user license **160** only when the user **150** was purchasing a new product license **154**. No password was required when the user **150** was merely playing a file **100** under an existing product license **154**. It would be well within the scope of the present invention to require that the password be entered by the user **150** whenever the user license **160** is accessed to validate a product license **154**. Alternatively, the password could be required just once each time the player software **152** is activated. Many possible combinations of features and elements are possible within the scope of the present invention, and therefore the scope thereof should be limited only by the following claims.

What is claimed is:

1. A method for distributing digital licenses over a computer network from a centralized database for a product comprising the steps of:

a) registering a vendor's product by storing a product identifier in conjunction with an encryption key and providing said encryption key to said vendor to use to encrypt said product; and

b) issuing a product license to a user, via a computer network, said product license including said product identifier information, but not said product itself, and a decryption key that mates with said encryption key.

2. A method for distributing digital licenses for a digital product comprising the steps of:

a) receiving a request, via a computer network, for product registration from a vendor, said request including a product name and a vendor identifier;

b) assigning a product identifier and an encryption key to said product and forwarding registration to vendor, said registration including a product identifier, said key and said vendor identifier;

c) receiving a request, via a computer network, from a user for a user license, said request including a user name and payment information;

d) assigning a user identifier to said user and forwarding a user license, via a computer network, to said user, said license including said user identifier;

e) receiving a request, via a computer network, for a product license from said user to use said product, said request including said user identifier and said product identifier;

f) issuing a product license to said user, via a computer network, said product license including a user identi-

fier, the product identifier and a decryption key that mates with said encryption key.

3. A method for obtaining a license for digitally distributed material, comprising the steps of:

a) downloading encrypted digital material carrying a product identifier;

b) purchasing a digital product license to use the material, said product license including a decryption key to decrypt the material and the product identifier.

4. A method according to claim 3, further comprising the steps of:

c) obtaining a user license, said user license including a user identifier;

and wherein said digital license includes said user identifier.

5. A method of obtaining a user license, comprising the steps of:

a) establishing a connection for data transmission between the user's computer and a license provider's computer;

b) transmitting via said data connection to the license provider a request for a user license including a user name and a system identifier that is unique to the user's computer;

c) receiving via data connection a user license from the license provider, said user license including a user identifier assigned by the license provider.

6. A method of obtaining a user license according to claim 5, further comprising the step of:

d) storing the user license on said user's computer.

7. A method of obtaining a user license according to claim 5, further comprising the steps of selecting and transmitting a password to the license provider and said user license incorporating said password.

8. A system for distributing user licenses, said system including:

a) a server connected to a user computer for data connection therebetween;

b) means for receiving from the user via said data connection a request for a user license, said request including a system identifier;

c) means for assigning and storing a unique user identifier in association with said system identifier d) a database for storing user license records, each said user license record including a user identifier and a system identifier.

9. A system for distributing user licenses according to claim 8, wherein each said user license record includes a user name.

10. A system for distributing user licenses according to claim 8, wherein each said user license record includes a password.

11. A system for distributing user licenses according to claim 8, wherein each said user license record includes the user's credit card number.

12. A system for distributing licenses for a product, comprising:

a) a product registration process for assigning to a product a unique product identifier and an encryption key;

b) a user registration process for assigning to a user a unique user identifier; and

c) a license distribution process coupled to the user registration process and to the product registration process for providing to the user a product license containing the user identifier and a decryption key that mates with said encryption key.

13. A system for managing rights in digital material comprising:

a) means for downloading a digital media product containing a product identifier;

b) means for storing downloaded digital material;

c) registry in device for playing the material, the registry containing a system identifier;

d) user license stored in the registry of the media player device, said user license containing a user identifier;

e) product license stored in memory that is operationally accessible to software running on the player, said product license containing the product identifier and the user identifier of the person authorized to play the material;

f) software for comparing the user identifier in the product license to the user identifier on the user license each time the software receives a request to play the product.

14. A method for managing licenses for digital data comprising:

a) assigning a vendor ID to owners of rights in digital data;

b) registering a collection of digital data as a product with a product database, and assigning a product ID and an encryption key to the product;

c) storing the product ID and the encryption key with the vendor ID in the product database;

d) encrypting at least a portion of the collection of digital data with the encryption key, and

e) storing the encrypted digital data and product ID in a product file.

15. The method of claim 14, further comprising:

f) assigning a user ID to a user who desires access to the encrypted digital data;

g) storing the user ID and a payment mechanism in a user database; and

h) storing the user ID and the payment mechanism electronically as a user license such that the user can access the user license without referring to the user database.

16. The method of claim 15, further comprising:

i) receiving a request from the user to purchase access to the product file, the request including the product ID and the user ID;

j) verifying that the product ID is found in the product database and the user ID is found in the user database;

k) using the payment mechanism stored in the user database with the user ID to secure payment;

l) storing the user ID and the product ID in a product license database; and

m) storing as a product license the encryption key associated with the product ID in the product database along with the user ID and the product ID, such that the user can access the product license without referring to the product license database.

17. The method of claim 16, further comprising:

n) in response to a request by the user to access the product file, searching for the product license containing the product ID of the product file;

o) comparing the user ID in the found product license with the user ID found in an accessible user license;

p) using the encryption key in the found product license to decrypt the encrypted digital data found in the product file.

18. The method of claim 17, wherein the product database, the user database, and the product license database are combined into a single database.

* * * * *