(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 618 230**

(51) Int. CI.:

H04L 29/06 (2006.01)
G06F 9/45 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **15.11.1999 PCT/US1999/27113**

(87) Fecha y número de publicación internacional: **25.05.2000 WO00030323**

(96) Fecha de presentación y número de la solicitud europea: **15.11.1999 E 99960380 (6)**

(97) Fecha y número de publicación de la concesión europea: **21.12.2016 EP 1131934**

(54) Título: **Procedimiento de ejecución de una aplicación sin estar instalada**

(30) Prioridad:

16.11.1998 US 108602 P
12.05.1999 US 310294
12.05.1999 US 311923
12.05.1999 US 310229
12.11.1999 US 439906

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
21.06.2017

(73) Titular/es:

MICROSOFT TECHNOLOGY LICENSING, LLC (100.0%)
One Microsoft Way
Redmond, WA 98052, US

(72) Inventor/es:

SCHMEIDLER, YONAH;
ATKINS, DEREK;
EICHIN, MARK, W. y
ROSTCHECK, DAVID, J.

(74) Agente/Representante:

CARPINTERO LÓPEZ, Mario

DESCRIPCIÓN

Procedimiento de ejecución de una aplicación sin estar instalada

Campo de la invención

La presente invención se refiere, en general, a un procedimiento y sistema para la distribución de datos a través de redes y, más en concreto a un sistema para distribuir contenido de soporte lógico ejecutable a través de redes de acceso de banda ancha de una forma segura que posibilita un abono a petición.

Antecedentes de la invención

La distribución a petición de aplicaciones de soporte lógico y tipos de datos multimedia tales como audio, vídeo, animación, etc. no ha resultado práctica hasta hace poco principalmente debido a las tasas a las que se transmiten los datos a través de las redes de comunicación. Se hace referencia a la tasa a la que se transmiten los datos, a los que se da formato de serie de bits, como bits por segundo (bps). Los primeros módems eran capaces de transmitir información a una tasa de aproximadamente 300 bits por segundo. A continuación de lo anterior, aumentaron las velocidades a las que los módems eran capaces de transmitir y de recibir datos. Con tales aumentos en la velocidad de módem, comenzaron a evolucionar la naturaleza de las topologías de red así como los tipos de datos que se transmiten a través de las redes. Con las velocidades de módem de 9600 bps y 1200 bps, las redes informáticas tales como Internet eran principalmente un entorno de texto ASCII con unos protocolos y mensajería de texto específicos. Los aumentos posteriores en la velocidad de módem posibilitaron el acceso a una información más compleja a través de Internet y otras redes informáticas. A pesar de que el paradigma de texto ASCII sigue existiendo en la porción de World Wide Web de Internet hoy en día, el entorno de banda ancha aumentada, más reciente, ha posibilitado la comunicación de tipos de datos multimedia y contenido más complejo.

De forma más reciente, la tecnología de banda ancha y módems por cable de alto rendimiento, con unas velocidades de conectividad de más de 1 millón de bps, están siendo distribuidos y ofrecidos por las empresas de cable, de teléfono, celular y por satélite por todo el mundo. Las redes de acceso de banda ancha actuales incluyen las redes Híbridas de Fibra - Coaxial (HFC, *Hybrid Fiber Coax*) de medio compartido de la industria del cable y las líneas de abonado digital (xDSL, *digital subscriber line*) de la industria telefónica.

Con la aparición de la tecnología de banda ancha y las redes de acceso de banda ancha, los abonados a servicios de red de acceso de banda ancha son en la actualidad capaces de acceder de forma remota a tipos de datos multimedia complejos y títulos de soporte lógico, previamente solo disponibles en Disco Compacto - Memoria de Solo Lectura (CD-ROM, *Compact Disc - Read Only Memory*) y Disco Versátil Digital (DVD, *Digital Versatile Disc*), a los que se hace referencia en lo sucesivo en el presente documento como “título o títulos”.

Hay, no obstante, factores que no son las tasas de datos que también han hecho poco práctica la distribución a petición de títulos. Un obstáculo de este tipo que evita la distribución a petición de contenidos, incluyendo soporte lógico y títulos multimedia, hasta la fecha, ha sido el requisito de tener el título cargado en el sistema informático local del abonado con el fin de ejecutar el título. Además, la copia generalizada o “pirateo” de contenido del título, y los riesgos de seguridad asociados, que están asociados con la distribución de copias plenamente habilitadas de títulos, ha convertido la distribución a petición en poco atractiva para las bibliotecas de contenidos y los publicadores de soporte lógico.

Por consiguiente, existe una necesidad de un procedimiento y sistema para la distribución a petición de contenido de soporte lógico ejecutable, que no requiera la instalación del contenido en el sistema informático local del abonado.

Existe una necesidad adicional de un procedimiento y sistema para distribuir contenido a los abonados de una forma a petición que proporcione seguridad para proteger el valor del contenido y que evite un uso no autorizado y el copiado del mismo.

Existe una necesidad adicional de un procedimiento y sistema en el que se pueda distribuir contenido a través de una red de acceso de banda ancha de una forma que cumpla los requisitos de latencia del contenido que se está ejecutando.

El documento EP 0 415 346 A2 divulga un procedimiento y unos medios para identificar de forma automática los medios que se usan con un sistema informático y para montar de forma automática y dinámica un sistema de archivos que reconoce los medios. Un controlador de sistema de archivos (FSD, *file system driver*) instalable es análogo en muchos sentidos a un controlador de dispositivo. Esto permite que un usuario cargue un controlador de dispositivo para un dispositivo no convencional y que cargue un controlador de sistema de archivos a partir de un volumen en ese dispositivo. Una vez que un FSD se ha instalado e inicializado, el núcleo se comunica con el mismo en términos de solicitudes lógicas de aperturas, lecturas, escrituras, búsquedas y cierres de archivos. El FSD traduce estas solicitudes usando unas tablas y estructuras de control que se hallan en el propio volumen a solicitudes de lecturas y escrituras de sectores para las que el mismo puede llamar a unos puntos de entrada de núcleo especiales que se denominan Ayudantes de Sistema de Archivos (FsHlp, *File System Helper*). El núcleo pasa las peticiones de E/S de sector al controlador de dispositivo apropiado y devuelve los resultados al FSD. Durante el

funcionamiento, el programa de aplicación 302 emite solicitudes lógicas de archivo al núcleo de sistema operativo 252 mediante una llamada a los puntos de entrada para la función deseada. Estas funciones pueden incluir solicitudes de apertura de archivos (DosOpen), de lectura de archivos (DosRead) y / o de escritura de archivos (DosWrite). El núcleo de sistema operativo 252 pasa estas solicitudes al controlador de sistema de archivos 254-258 apropiado para el volumen particular que contiene el archivo. El controlador de sistema de archivos instalable apropiado traduce entonces la solicitud lógica de archivo a solicitudes de lecturas o escrituras de sectores lógicos de los medios designados y llama a un ayudante de sistema de archivos de núcleo de sistema operativo 308 para pasar estas solicitudes al controlador de dispositivo 306 apropiado.

Por lo tanto, el objeto de la presente invención es la provisión de un procedimiento mejorado de ejecución de una aplicación en un sistema informático local sin que la aplicación esté instalada en el sistema informático local, y un aparato correspondiente.

El presente objeto se soluciona mediante la materia objeto de las reivindicaciones independientes.

Mediante las reivindicaciones dependientes se definen realizaciones preferidas.

título identificado de forma única por el testigo; y (d) suministrar al cliente al menos una porción del título identificado por el testigo.

Breve descripción de los dibujos

Las características, objetos y ventajas anteriores, así como otros, de la invención, se entenderán mejor al hacer referencia a la siguiente descripción detallada en conjunción con los dibujos adjuntos en los que:

la figura 1 es un diagrama de bloques de un sistema informático conveniente para su uso con la presente invención;
la figura 2A es un diagrama de bloques conceptual de una red de banda ancha en la que se puede implementar el sistema de distribución de contenido seguro; H
la figura 2B es un diagrama de bloques conceptual que ilustra los elementos del sistema y la interacción con otros elementos de red;
la figura 3A es un diagrama de bloques conceptual del cliente de SCDP;
la figura 3B es un diagrama de bloques conceptual del módulo de iniciador del cliente de SCDP de la figura 3A;
la figura 3C es un diagrama de bloques conceptual del módulo de VxD de ARFS del cliente de SCDP de la figura 3A;
la figura 3D es un diagrama de bloques conceptual del módulo de VxD de RAFT del cliente de SCDP de la figura 3D;
las figuras 4A-B forman colectivamente un diagrama de flujo que ilustra el proceso de abono a contenido y de inicio de un título;
las figuras 5A-C forman colectivamente un diagrama de flujo que ilustra las etapas de proceso que son realizadas por el cliente de SCDP;
las figuras 6 es un diagrama de flujo que ilustra el proceso que se ejecuta mediante los componentes de cliente de SCDP de acuerdo con la presente invención;
la figura 7A es un diagrama conceptual del servidor de CAS de la figura 2;
la figura 7B es un diagrama de flujo que ilustra el proceso que se ejecuta mediante el servidor CAS;
la figura 8 es un diagrama conceptual de un testigo de RAFT;
la figura 9 es un diagrama conceptual de una cadena de inicio;
la figura 10 es un diagrama conceptual del servidor de RAFT de la figura 2;
la figura 11 es un diagrama conceptual de un encabezado de paquete de RAFT;
la figura 12 es un diagrama conceptual de un paquete de datos de BRIQ;
la figura 13 es un diagrama de bloques conceptual de un activador; y
la figura 14 es un diagrama de bloques conceptual de un servicio de comercio electrónico.

Descripción detallada

La figura 1 ilustra la arquitectura de sistema para un sistema informático 100 tal como una estación de trabajo SparcStation 5 de Sun, facilitada a nivel comercial por Sun Microsystems de Palo Alto, CA, o una estación de trabajo RS/6000 de IBM o un PC Aptiva de IBM, ambos facilitados a nivel comercial por International Business Machines Corp. de Armonk, N. Y., en el que se puede implementar la invención. El sistema informático a modo de ejemplo de la figura 1 es solo para fines descriptivos. A pesar de que la descripción puede hacer referencia a expresiones comúnmente usadas al describir sistemas informáticos particulares, la descripción y los conceptos son igualmente de aplicación a otros sistemas, incluyendo sistemas que tienen arquitecturas diferentes de las de la figura 1.

El sistema informático 100 incluye una unidad central de procesamiento (CPU, *central processing unit*) 105, que se puede implementar con un microprocesador convencional, una memoria de acceso aleatorio (RAM, *random access memory*) 110 para un almacenamiento temporal de información, y una memoria de solo lectura (ROM, *read only memory*) 115 para un almacenamiento permanente de información. Se proporciona un controlador de memoria 120 para controlar la RAM 110.

Un bus 130 interconecta los componentes del sistema informático 100. Se proporciona un controlador de bus 125 para controlar el bus 130. Un controlador de interrupción 135 se usa para recibir y procesar varias señales de interrupción a partir de los componentes del sistema.

El almacenamiento masivo se puede proporcionar mediante el disquete 142, el CD ROM 147 o la unidad de disco duro 152. Los datos y el soporte lógico se pueden intercambiar con el sistema informático 100 por medio de medios extraíbles tales como el disquete 142 y el CD ROM 147. El disquete 142 es insertable en la unidad de disquete 141 que, a su vez, está conectada con el bus 30 mediante un controlador 140. De forma similar, el CD ROM 147 es insertable en la unidad de CD ROM 146 que, a su vez, está conectada con el bus 130 mediante el controlador 145. El disco duro 152 es parte de una unidad de disco fijo 151 que está conectada con el bus 130 mediante el controlador 150.

La entrada de usuario al sistema informático 100 se puede proporcionar mediante un número de dispositivos. Por ejemplo, un teclado 156 y un ratón 157 están conectados con el bus 130 mediante el controlador 155. Un transductor de audio 196, que puede actuar tanto como un micrófono como un altavoz, está conectado con el bus 130 mediante el controlador de audio 197, tal como se ilustra. Será obvio a los razonablemente expertos en la materia que otros dispositivos de entrada, tales como un lápiz y / o un tabloide, se pueden conectar con el bus 130 y un controlador y un soporte lógico apropiados, según se requiera. El controlador de DMA 160 se proporciona para realizar un acceso directo de memoria a la RAM 110. Una presentación visual se genera mediante el controlador de vídeo 165 que controla la pantalla de vídeo 170. El sistema informático 100 también incluye un adaptador de comunicaciones 190 que permite que el sistema esté interconectado con una red de área local (LAN, *local area network*) o una red de área extensa (WAN, *wide area network*), que se ilustra de manera esquemática por medio del bus 191 y la red 195.

El funcionamiento del sistema informático 100 en general se controla y se coordina mediante el soporte lógico de sistema operativo, tales como Windows 95 o Windows NT®, facilitados a nivel comercial por Microsoft Corp., Redmond, WA. El sistema operativo controla la atribución de los recursos de sistema y realiza tareas tales como procesamiento de programación, gestión de memoria, funciones de red y servicios de E/S, entre otras cosas. En particular, un sistema operativo que reside en memoria de sistema y que se está ejecutando en la CPU 105 coordina el funcionamiento de los otros elementos del sistema informático 100. La presente invención se puede implementar con cualquier número de sistemas operativos disponibles en el mercado incluyendo OS/2®, UNIX®, Linux y Solaris®, entre otros. Una o más aplicaciones de navegadores tales como Netscape Navigator, versión 2.0 y posteriores a esa, facilitada a nivel comercial por Netscape Communications Corporation e Internet Explorer, versión 1.0 y posteriores a esa, facilitada a nivel comercial por Microsoft Corporation, Redmond, Washington, se pueden ejecutar bajo el control del sistema operativo.

Visión general del sistema de SCDP

La figura 2A ilustra de forma conceptual los componentes principales de un sistema de Plataforma de Entrega de Contenido Segura (SCDP, *Secure Content Delivery Platform*) 200, así como otros elementos en un entorno de red de banda ancha, siendo tal entorno solo para fines a modo de ejemplo y no se ha de considerar limitante. Los elementos que se ilustran en la figura 2A son para facilitar y comprender la invención. No todos los elementos que se ilustran en la figura 2A o que se describen en el presente documento son necesarios para la implementación o el funcionamiento de la invención. Tal como se ilustra en la figura 2A, el sistema de SCDP 200 comprende un Servidor de Acceso Condicional (CAS, *Conditional Access Server*) 210, una base de datos de CAS 212 asociada, un Servidor de Transferencia de Archivos de Acceso Aleatorio (RAFT, *Random Access File Transfer Server*) 206, una base de datos de RAFT 208 y el cliente de SCDP 216.

Además del servidor de CAS 210, el servidor de RAFT 206 y el Cliente de SCDP 216, la presente invención contempla el uso de un escaparate electrónico virtual 215 y un servidor de comercio electrónico 202. El servidor de comercio electrónico 202 tiene una base de datos de facturación 204 adjunta. El escaparate electrónico 215 tiene una base de datos 213 adjunta. En la realización ilustrativa, los servidores 202, 210 y 215 están conectados a través de una red de área local (LAN, *local area network*) privada y segura, tal como una red de Ethernet local. La LAN, a su vez, está conectada con una topología global de red informática, que se ilustra como la nube de Internet 240 en la figura 2A, mediante un proveedor de servicios de Internet (ISP, *Internet service provider*) 230. Cualquier número de proveedores de servicio de acceso a Internet disponibles en el mercado tales como MCI WorldCom, AT&T, America OnLine, etc. se pueden usar como el ISP 230. En la realización ilustrativa, a pesar de que los servidores 202, 210 y 215 se ilustran como estando conectados a través de una red de área local privada, será obvio a los expertos en la materia que tales servidores se pueden acoplar operativamente a través de otras redes no privadas, tales como Internet. Además, el servidor de comercio electrónico 202 se puede acoplar con un servidor de procesamiento de crédito de una institución financiera o bancaria (que no se muestra) para ayudar en el procesamiento de tarjetas de crédito y / u otros tipos de transacciones.

Haciendo referencia de nuevo a la figura 2A, uno o más PC de cliente que tienen una arquitectura similar a la de la figura 1, están conectados con el sistema de SCDP 200 a través de una red de acceso de banda ancha 203 y el proveedor de cable 207. En la realización ilustrativa, un módem por cable (CM, *cable modem*) conecta con el PC anfitrión en el que se está ejecutando el cliente de SCDP. A su vez, una pluralidad de módems por cable están

acoplados con un nodo de cable por medio de una conexión de alta frecuencia. Por lo general, tantos como 1.000 PC anfitriones se pueden conectar a un nodo de cable a través de módems por cable y conexiones de alta frecuencia apropiados. Cada nodo de cable, a su vez, está conectado a través de un sistema de finalización de módem por cable (CMTS, *cable modem termination system*). Una pluralidad de sistemas de finalización de módem por cable están acoplados con una cabecera de finalización. Una pluralidad de cabeceras interconectadas comprenden la red troncal de la red de acceso de banda ancha. Por lo general, las cabeceras de cable están ubicadas en las instalaciones de la compañía de cable y pueden incluir un terminal de datos de anfitrión que está conectado con una red de Protocolo de Internet (IP, *Internet Protocol*) a través de una línea de T1 u otra conexión. La línea de T1, a su vez, se puede conectar a Internet a través de un proveedor de servicios de Internet (ISP, *Internet Service Provider*) 230. El servidor de RAFT 206 y su base de datos 208 adjunta están acoplados con la red de acceso de banda ancha 203 entre el proveedor de servicios de Internet 230 y la cabecera o instalación de finalización de datos de anfitrión que proporciona la compañía de cable. De esta forma, el servidor de RAFT 206, a pesar de ser parte del sistema de SCDP 200, está ubicado de forma remota con respecto al CAS 210, el servidor de comercio electrónico 202 y el escaparate electrónico virtual 215. El sistema de finalización de módem por cable 209 convierte datos de alta frecuencia a partir de una infraestructura de cable en formato de Protocolo de Internet usando la Norma de la Industria de Servicios de Datos a través de Cable (DOCSIS, *Data Over Cable Service Industry Standard*).

Como alternativa, un PC de cliente se puede conectar con el sistema de SCDP 200 por medio de un servicio de línea de abonado digital (DSL, *digital subscriber line*), tal como se ilustra en la figura 2A. En esta configuración, un ordenador anfitrión en el que se está ejecutando el cliente de SCDP está acoplado con un conmutador de compañía de teléfono por medio de un módem de DSL y la infraestructura de red pública de telefonía conmutada existente.

La construcción de las redes de abonado de DSL y de las redes de acceso de banda ancha se conoce en la técnica y se usan en la actualidad por las compañías de cable y las compañías de teléfono de forma exhaustiva y no se describirán con detalle adicional en el presente caso por razones de brevedad. Por consiguiente, no todos los elementos de los sistemas que se han descrito en lo que antecede se ilustran en la figura 2A.

Proceso de abono

La figura 2B ilustra de forma conceptual la interacción de los componentes dentro del sistema de SCDP 200. El diagrama de flujo de las figuras 4A-B en conjunción con el diagrama de bloques conceptual de la figura 2B ilustra las etapas de procedimiento que son realizadas por el sistema de SCDP 200 durante los procesos de abono y de inicio.

Un usuario que está equipado con el cliente de SCDP 216 que se está ejecutando en un PC y un navegador de HTML por ejemplo, Netscape Navigator o Microsoft Internet Explorer, selecciona un título a partir del escaparate electrónico virtual 215, tal como se ilustra por medio de la etapa 401. En el escaparate electrónico 215, cada título disponible se publica como una oferta digital que está incrustada dentro de un Localizador Universal de Recursos (URL, *Universal Resource Locator*). La oferta digital contiene una información que identifica el título y el tipo de compra seleccionados. La selección de la oferta digital dirige el navegador del abonado al extremo frontal de HTTP 202A del servidor de comercio electrónico 202, tal como se ilustra por medio de la etapa 402. El usuario negocia con el servidor de comercio electrónico 202 para una compra basándose en la información en el URL de oferta digital, tal como se ilustra por medio de la etapa 403. Por lo general, la negociación puede comportar el registro de usuario y la provisión de información de crédito.

El servidor de comercio electrónico genera una cadena de inicio, que contiene la información que identifica y que autoriza la compra, incluyendo un Nombre Universal de Recurso (URN, *Universal Resource Name*) que identifica de forma única el contenido deseado, tal como se ilustra por medio de la etapa 404A. El formato y la descripción del URN y la cadena de inicio se describen en lo sucesivo en el presente documento. La cadena de inicio se firma digitalmente mediante el CAS 210 y se proporciona al servicio de comercio electrónico 202 para la distribución al cliente de SCDP 216, tal como se ilustra por medio de la etapa 404B.

La cadena de inicio se encapsula con un encabezado de MIME (Extensión de Correo de Internet Multipropósito, *Multipurpose Internet Mail Extension*). Cuando la cadena de inicio es recibida por el navegador 224 del cliente de SCDP, el tipo de MIME que está asociado con la cadena de inicio está ubicado en una entrada de registro, lo que da como resultado la invocación del módulo de iniciador 220 dentro del cliente de SCDP 216, tal como se ilustra por medio de la etapa 405. El iniciador 220 establece una conexión de RPC segura con el CAS 210 y solicita que el CAS proporcione un URL para el URN especificado, es decir, una conversión de URN a URL, tal como se ilustra por medio de la etapa 406A. El URL identifica la ubicación de los correspondientes datos de BRIQ. El CAS 210 reenvía el URL correspondiente al iniciador 220. Una vez que el iniciador ha identificado la ubicación de los correspondientes datos de BRIQ, el iniciador envía una solicitud de compra al CAS, incluyendo la solicitud de compra la cadena de inicio, tal como se ilustra por medio de la etapa 406B.

El CAS verifica la firma de la cadena de inicio y, entonces, devuelve un activador y un testigo de autorización de RAFT al iniciador, tal como se ilustra por medio de la etapa 407. El activador y el testigo de autorización se describen en lo sucesivo en el presente documento con mayor detalle. El testigo de autorización se puede incrustar en la práctica dentro del activador. A continuación, el iniciador inicia el título al pasar el activador al VxD de ARFSD

218, tal como se ilustra por medio de la etapa 408. El VxD de ARFSD ejecuta el activador que pasa el testigo de autorización de RAFT al VxD de RAFT 222. El VxD de RAFT abre el URL y lee el encabezado, tal como se ilustra por medio de la etapa 409. El VxD de RAFT envía el testigo de autorización inicial al servidor de RAFT, tal como se ilustra por medio de la etapa 410. El VxD de RAFT 222 comienza a leer contenido a partir del servidor de RAFT 206, pasando el contenido recibido de vuelta al VxD de ARFSD 218, tal como se ilustra por medio de la etapa 411. El VxD de ARFSD usa el activador para descifrar y descomprimir el contenido en forma de datos de BRIQ, y realizar una comprobación de integridad sobre los bloques de datos descifrados, tal como se ilustra por medio de la etapa 412.

A continuación de lo anterior, el sistema operativo ejecuta el título, por medio del sistema de archivos local que es presentado por un VxD de ARFSD, tal como se ilustra por medio de la etapa 413. De forma periódica, el activador solicita al iniciador 220 que pida al CAS 210 que renueve el activador y el testigo de autorización de RAFT. Tras la primera de tales solicitudes, el CAS publica la compra en el servidor de comercio electrónico 202 para la resolución de transacciones, tal como se ilustra por medio de la etapa 414. El tiempo de vida del primer activador puede ser del orden de minutos. La renovación de activador con éxito después del primer tiempo de espera sirve como una indicación de que el título se está ejecutando con éxito.

Habiendo proporcionado una visión general de los componentes del sistema y de su interacción, una descripción más detallada del sistema de distribución de contenido seguro 200 de la presente invención y los procesos que se realizan de ese modo se exponen con referencia a las figuras 3A-14 y sus explicaciones adjuntas.

Cliente de SCDP

Haciendo referencia a la figura 3A, se ilustra un diagrama de bloques conceptual del cliente de SCDP 216. El cliente de SCDP 216 permite que los usuarios ejecuten títulos codificados por BRIQ en un PC anfitrión. Tal como se ilustra en la figura 3A, el cliente de SCDP comprende el iniciador 220, el VxD de Controlador de Sistema de Archivos de Arepa (VxD de ARFSD, *Arepa File System Driver*) 218, y el VxD de cliente de RAFT 222. El cliente de SCDP 216 se puede implementar como una aplicación ejecutable en el sistema operativo 219, por ejemplo, una aplicación de Windows (R) en la realización ilustrada. El sistema operativo 219 es ejecutable encima de una arquitectura de PC, tal como un PC de IBM u otra arquitectura informática, tal como se describe con referencia a la figura 1. Además del cliente de SCDP 216, un navegador 217, por lo general un navegador de HTML tal como NetScape Navigator o Microsoft Explorer, también se puede estar ejecutando bajo el control del sistema operativo 219. El iniciador 220, el VxD de ARFSD 218 y el VxD de RAFT 222 se describen con mayor detalle con referencia a las figuras 3B-D, respectivamente.

La figura 3B ilustra de forma conceptual un diagrama de bloques de los módulos de lógica de programa que comprenden el iniciador 220 del cliente de SCDP 216. En concreto, el iniciador 220 comprende un módulo de control 300, una biblioteca de RPC de CAS 302, una biblioteca de comunicación de VxD de ARFSD 304 y una interfaz de usuario 306. En la realización ilustrativa, el iniciador 220 se puede implementar como una aplicación de Windows que contiene una lógica que coordina toda la comunicación entre el cliente de SCDP y el CAS 204. El iniciador 220 es invocado por el navegador web 217 del cliente, tras la compleción de la negociación de compra con el sistema de comercio electrónico 202. El sistema de comercio electrónico entrega al navegador web del cliente una cadena de inicio con el tipo de MIME que está asociado con el iniciador. Además, el iniciador gestiona todas las comunicaciones con el CAS, incluyendo 1) obtener del CAS la dirección del servidor de RAFT y el nombre de ruta de BRIQ que se corresponden con el título seleccionado; 2) obtener del CAS un activador y un testigo de autorización de RAFT necesarios para recuperar datos de BRIQ del servidor de RAFT y para descifrar los datos recuperados; y 3) pedir al CAS que renueve el testigo de autorización de RAFT y el activador.

Para facilitar la comunicación entre el servidor CAS 206 y el módulo de VxD de ARFSD 218, el iniciador 220 incluye la biblioteca de RPC de CAS 302, que se puede implementar como una serie de objetos o código de programa que generan y reciben comunicaciones a / de el servidor CAS 206 a través de una biblioteca de llamadas a procedimiento remoto (RPC, *remote procedure call*). Una biblioteca de RPC de este tipo conveniente para su uso como el módulo 302 es el Producto de RPC Segura de Noblenet facilitado a nivel comercial por Noblenet, Inc. Opcionalmente, un producto de transporte de red, tal como los que siguen la norma de Biblioteca de Zócalo Seguro (SSL, *Secure Socket Library*) publicada por Netscape Communications Corporation, se puede usar para transportar las llamadas RPC a través de la red y, por lo tanto, potencian adicionalmente la seguridad de las transmisiones a / desde el cliente de SCDP en el sistema de la presente invención. Una biblioteca de comunicación también se utiliza para las comunicaciones entre el módulo de iniciador 220 y el módulo de VxD de ARFSD 218 y entre el módulo de VxD de ARFSD 218 y el VxD de RAFT 222. Tal biblioteca de nuevo incluye código u objetos necesarios para comunicar datos entre el iniciador 220 y el VxD 218. Por ejemplo, tal como se describe con mayor detalle en lo sucesivo en el presente documento, una información seleccionada a partir del encabezado de BRIQ 1202 del BRIQ 1200, tal como se ilustra en la figura 12, se lee mediante el módulo de control 300 y se suministra al VxD 218 a través de la biblioteca de comunicación 304, durante la ejecución de un título.

Tras la invocación del iniciador 220, una interfaz gráfica de usuario (GUI, *graphic user interface*) se presenta a un usuario a través de la interfaz de usuario 306. En la realización ilustrativa, la interfaz de usuario 306 incluye la lógica de programa y / o los objetos apropiados necesarios para interactuar con la interfaz de sistema operativo y con las

interfaces de programación de aplicaciones (API, *Application Program Interface*) contenidas dentro del sistema operativo Windows con el fin de mostrar ventanas, presentar una información gráfica dentro de tales ventanas y recibir instrucciones procedentes de un usuario por medio de un teclado, un ratón u otro dispositivo apuntador, encontrándose tal interfaz de usuario bien dentro del ámbito de los razonablemente expertos en la materia. A través de esta GUI, el usuario puede establecer preferencias de usuario, por ejemplo, tamaño de memoria caché de disco, y recibir notificaciones de condiciones de error.

El módulo de control 300 se puede implementar con el código o los objetos apropiados para llevar a cabo el algoritmo necesario para iniciar un título y continuar las comunicaciones entre el cliente de SCDP 200 y el CAS 210 y el servidor de RAFT 206. Más en concreto, los algoritmos que se ejecutan mediante el módulo de control 300 se ilustran con mayor detalle en las figuras 4A-6 y sus descripciones adjuntas.

La figura 3C es un diagrama de bloques conceptual del VxD de ARFSD 304 de la figura 3B. El VxD 304 comprende un intérprete de códigos de bytes 308, un módulo de control 310 y una biblioteca de comunicación de VxD de ARFSD 312. El VxD de ARFSD 304 es un controlador de dispositivo virtual posibilita que el sistema operativo lea los datos de BRIQ como un sistema de archivos local. El VxD de ARFSD 304 descomprime y descifra datos de BRIQ. Además, el VxD de ARFSD mantiene la abstracción de instalación, por ejemplo, suministrando información de registro a Windows. El VxD de ARFSD implementa entradas de registro dinámicas mediante la intercepción de todas las llamadas de acceso a registro de sistema operativo y, entonces, la simulación de las entradas de registro que están asociadas con el título en ejecución, pero no guardadas en disco.

Activadores y el intérprete de códigos de bytes

Tal como se ha descrito en lo que antecede, el activador 228 se ejecuta en el intérprete de códigos de bytes 308 materializado en el VxD de ARFSD 304. El activador representa una porción del soporte lógico de cliente de SCDP que se obtiene del CAS 210, y que emplea el VxD de ARFSD 304 para descifrar datos de BRIQ. La forma y el contenido del activador tal como se describe con mayor detalle con referencia a la figura 13. Los activadores implementan un mecanismo de conexión persistente que requiere a los activadores que pidan de forma periódica activadores de sustitución al CAS 210. Por lo tanto, la comunicación con el CAS se ha de mantener con el fin de continuar la ejecución de un título. En la realización ilustrativa, el mecanismo de conexión persistente dentro del activador 228 se puede implementar como una cadena numérica o tal como se describe por lo demás con referencia a la figura 13.

El activador se implementa como un objeto de código de bytes dinámico que se puede ejecutar dentro del VxD de ARFSD 304. El CAS genera activadores a través de una llamada a las rutinas de generación de activadores que pueden residir en una biblioteca externa, tal como se ha descrito en lo que antecede con referencia al módulo de fábrica de activadores 710 de la figura 7. El testigo de RAFT, que se ha analizado en lo que antecede, se empaqueta con el activador. Con el tiempo, el activador expirará, después de lo cual el cliente de SCDP 216 ha de llamar al CAS y solicitar un nuevo activador. La vida del activador se determina mediante los valores de datos de tiempo de inicio y de tiempo de fin que están contenidos dentro de la porción de testigo del activador.

El sistema de SCDP 200 usa activadores para proteger la liberación de material criptográfico al cliente de SCDP 216. Un activador se puede implementar como un fragmento de código de bytes ofuscado que se ejecuta en el interior del VxD de ARFS 304 y posibilita el descifrado de un BRIQ. Una vez que el activador se ha descargado, este puede hacer RPC adicionales al CAS 210 para finalizar la distribución del material de generación de claves. La ofuscación de código dentro del activador puede proteger frente a la extracción de las claves.

La implementación ilustrativa de activadores también utiliza una ejecución remota para proteger las claves en el activador. La ejecución remota hace que el activador esté incompleto, es decir, da suficiente información al activador para continuar el funcionamiento durante un periodo de tiempo limitado y, entonces, requiere que el activador solicite código o datos adicionales. El intérprete de códigos de bytes 308 dentro del VxD de ARFSD 304 comprende una lógica de programa, código u objetos que extraen los datos de clave criptográfica del activador. En la realización ilustrativa, el activador puede tener el formato y el contenido tal como se describe con referencia a la figura 13.

En realizaciones alternativas, que utilizan unas implementaciones de activador más sofisticadas en las que el activador contiene códigos de bytes ofuscados, el intérprete de códigos de bytes 308 dentro del VxD de ARFSD 304 se puede implementar con un conjunto de instrucciones rico, para aumentar las oportunidades de ofuscación simple. Obsérvese que no existe límite tradicional alguno de conjuntos de instrucciones de microprocesador implementados por soporte físico y, por lo tanto, se pueden usar muchos bits para modos de direccionamiento y formatos de instrucciones. La complejidad y confidencialidad de tal conjunto de instrucciones permite la entrega segura de contenido dentro del sistema de SCDP. Debido a que el código de bytes se ejecuta en el interior de un VxD, el intérprete de códigos de bytes 308 puede llamar a interfaces exportadas de otros VxD pero no es necesario que llame a funciones de WIN32 a partir del sistema operativo o que maneje DLL.

El intérprete de códigos de bytes 308, en la realización ilustrativa, se implementa como una máquina virtual que tiene el código y / o la lógica de programa apropiados necesarios para interpretar y ejecutar los códigos de bytes que están contenidos dentro de un activador que se recibe del CAS 210. Una máquina virtual de este tipo incluye las

rutas apropiadas para interpretar el código de bytes o códigos de bytes, almacenar cualquier dato temporal a partir del flujo de códigos de bytes y ejecutar los procesos identificados por el código de bytes o códigos de bytes. La implementación específica del intérprete de códigos de bytes 308, por lo tanto, depende del conjunto de códigos de bytes ejecutable por el intérprete. Por ejemplo, el intérprete de códigos de bytes 308 puede implementar un número de características específicas con el fin de dar cabida al tipo de código que contienen los activadores, incluyendo cualquiera de los siguientes:

- Operadores Bit a Bit -- funciones de desplazamiento, rotación y “extracción de bits” que son útiles para las rutinas criptográficas y de serialización;
- Eval -- “llamada a datos” explícita que deja que el intérprete de códigos de bytes interprete códigos de bytes descargados o modificados, evitando de ese modo la separación de código y datos e indicadores correspondientes y protección de datos; o
- Primitivas de interconexión -- El intérprete de códigos de bytes de cliente de SCDP llama a funciones en otros VxD directamente, incluyendo la serialización de argumentos y la internalización de un conjunto previamente definido particular de tipos de C. Tanto el cliente de SCDP como el CAS utilizan primitivas de Interconexión de Flujo Seguro, por ejemplo, enlaces para extraer datos de conexión, en particular datos de autenticación, del flujo al que está unido el activador o la técnica.

Será obvio a los expertos en la materia que el intérprete de códigos de bytes 308 también se puede implementar como una máquina física. En la implementación de activador más simple que se describe en el presente documento, los códigos de bytes son opcionales. Por consiguiente, el intérprete de códigos de bytes 308 también puede ser opcional.

La biblioteca de comunicación 312 se utiliza para las comunicaciones entre el módulo de VxD de ARFSD 218 y el módulo de VxD de RAFT 222. Tal biblioteca es similar a la biblioteca de comunicación 304 de la figura 3B y facilita las comunicaciones entre los VxD 218 y 220.

El módulo de control 310 incluye la lógica de programa o código necesario para llevar a cabo el algoritmo necesario para realizar la abstracción de instalación, ejecutar un título y renovar un testigo de RAFT. Más en concreto, los algoritmos que se ejecutan mediante el módulo de control 310 se ilustran con mayor detalle en las figuras 4A-6 y sus explicaciones adjuntas.

La figura 3D ilustra de forma conceptual un diagrama de bloques de los componentes que comprenden el VxD de RAFT 222 de cliente de SCDP 216. En concreto, el VxD 222 comprende una biblioteca de RPC de RAFT 316, una lógica de almacenamiento en memoria caché 318 y un módulo de control 320. La biblioteca de RPC 316 contiene el código apropiado y / u objetos que implementan la capa de RPC de lado de cliente del protocolo de RAFT, que se describe con mayor detalle en el presente documento. Tal lógica de programa se utiliza para comunicarse con el servidor de RAFT 206 utilizando uno de los mensajes de protocolo de RAFT. En concreto, el módulo 316 contiene la lógica necesaria para adjuntar un encabezado de paquete de RAFT, tal como se describe con referencia a la figura 11, a cada mensaje de protocolo de RAFT y para responder con el apropiado de los mensajes de protocolo de RAFT. La lógica de almacenamiento en memoria caché 318 contiene el código apropiado para realizar un almacenamiento en memoria caché de los BRIQ, o porciones de los mismos, que se recuperan del servidor de RAFT 206 usando el protocolo de RAFT. Las porciones de los BRIQ que se almacenan en memoria caché mediante el módulo 218 se pueden almacenar en una porción de memoria temporal en el PC anfitrión en el que se está ejecutando el cliente de SCDP 216. La técnica de almacenamiento en memoria caché particular y su lógica asociada se pueden implementar de acuerdo con cualquier número de una pluralidad de algoritmos de almacenamiento en memoria caché conocidos, fácilmente dentro de la comprensión de los razonablemente expertos en la materia. El módulo de control 320 se puede implementar para incluir la lógica de programa, código y / u objetos necesarios para supervisar las funciones que se han descrito en lo que antecede con respecto a los módulos 316 y 318 y para ejecutar las etapas de procedimiento que se describen con referencia a las figuras 4A-6.

Ejecución de un título

El diagrama de flujo de las figuras 5A-C ilustra las etapas de procedimiento que son realizadas por el cliente de SCDP 216 durante una ejecución de títulos típica.

Tal como se ha expuesto previamente, cuando una cadena de inicio es recibida por el navegador 224 del cliente de SCDP, el tipo de Extensión de Correo de Internet Multipropósito (MIME, *Multipurpose Internet Mail Extension*) que está asociado con la cadena de inicio está ubicado en una entrada de registro, lo que da como resultado la invocación del módulo de iniciador 220 dentro del cliente de SCDP 216. Tras la invocación, el iniciador 220 extrae el Nombre Universal de Recurso (URN, *Universal Resource Name*) a partir de la cadena de inicio y solicita al CAS 210 que realice una conversión de URN a URL, tal como se ilustra por medio de la etapa 6. El URN es un identificador único de un título dentro de un BRIQ. El formato de URN convencional es tal como sigue: urn:arepa://proveedor/ruta/nombretítulo [n.º de versión]

En el URN, no es necesario que la ruta al título se corresponda exactamente con la ubicación actual del título en el servidor de almacenamiento del proveedor. La ruta es una comodidad de categorización, y no es necesaria. El

número de versión del título es opcional, y puede estar separado del nombre del título por un signo de libra. El nombre de proveedor se puede registrar con una autoridad central con el fin de asegurar la unicidad.

El Localizador Universal de Recursos (URL, *Universal Resource Locator*) identifica la ubicación actual de un BRIQ en un servidor de almacenamiento de RAFT. El formato convencional de URL es tal como sigue: raft://nombreanfitrión/ruta/nombreBRIQ.brq.

En un URL, la ruta se ha de corresponder exactamente con la ubicación actual del BRIQ en el servidor de almacenamiento de RAFT.

Las figuras 5A-C forman colectivamente un diagrama de flujo que ilustra las etapas de proceso que son realizadas por el cliente de SCDP y los módulos que están contenidos en el mismo durante el proceso de abono y de ejecución de títulos. Haciendo referencia también a los elementos de la figura 2B, un usuario del ordenador anfitrión en el que se ejecuta el cliente de SCDP utiliza un navegador web 224 para seleccionar el título deseado a partir del escaparate electrónico virtual 215. El escaparate electrónico 215 devuelve una oferta digital al navegador web, con la oferta digital el usuario negocia una compra con el servidor de comercio electrónico 202. El servidor de comercio electrónico transmite una cadena de inicio no firmada de vuelta al navegador web a través de la red. La cadena de inicio se encapsula con un encabezado de MIME. Cuando la cadena de inicio es recibida por el navegador, el tipo de MIME que está asociado con la cadena de inicio está ubicado en una entrada de registro de sistema de archivos dando como resultado la invocación del módulo de iniciador 220 del cliente de SCDP, tal como se ilustra por medio de la etapa 502. El módulo de iniciador 220 extrae el valor de URN a partir de la cadena de inicio y transmite el valor de URN al servidor CAS 210, tal como se ilustra en la etapa de procedimiento 504. Las comunicaciones entre el iniciador 220 y el CAS 210 se establecen a través de una conexión de RPC segura. El CAS 210 proporciona una conversión de URN a URL y transmite el URL correspondiente al cliente de SCDP. Una vez que el URL se ha recibido, tal como se indica por medio de la etapa de decisión 506, el iniciador 220 pasa una solicitud para leer el encabezado de URL al VxD de ARFSD 218, que, a su vez, pasa la solicitud al VxD de RAFT 222. El VxD 222 transmite la solicitud usando el protocolo de RAFT al servidor de RAFT 206. El servidor de RAFT 206 abre el URL y lee la información de encabezado. La información de encabezado se pasa entonces de vuelta al VxD de RAFT 222, al VxD de ARFS 218 y al iniciador 220. La totalidad de este proceso se representa por medio de la etapa de procedimiento 508 de la figura 5A. A continuación, el módulo de iniciador 220 utiliza el contenido del encabezado para realizar los requisitos de realización de pruebas de aplicación del sistema anfitrión, tal como se ilustra por medio de la etapa de procedimiento 510.

A continuación de la compleción de los requisitos de puesta a prueba de sistema, el módulo de iniciador 220 transmite una solicitud de autorización de compra, por medio de una conexión de RPC segura, al CAS 210, tal como se ilustra en la etapa de procedimiento 512. En respuesta a la solicitud de autorización de compra, el CAS 210 genera un activador, incluyendo un testigo de RAFT, que se transmite a través de la conexión de RPC segura al cliente de SCDP 216. Tras la recepción del activador, tal como se indica por medio de la etapa de decisión 514, el módulo de iniciador 220 instala el activador y el testigo de RAFT, tal como se indica por medio de la etapa de procedimiento 516. El activador se instala en el VxD de ARFSD 218, que, a su vez, carga el testigo de RAFT en el VxD de RAFT, tal como se ilustra por medio de las etapas de procedimiento 516 y 518, respectivamente. El VxD de RAFT 222 transmite entonces el testigo de RAFT al servidor de RAFT 206 usando una de las instrucciones apropiadas del protocolo de RAFT, tal como se ilustra por medio de la etapa de procedimiento 520. A continuación, el VxD de ARFSD 218, a través de las comunicaciones con el VxD 222 lee el campo de superbloque a partir del BRIQ que está ubicado en el servidor de RAFT 206, tal como se ilustra por medio de la etapa de procedimiento 522, y verifica un número mágico en el superbloque, tal como se ilustra por medio de la etapa de procedimiento 524. El número mágico en el BRIQ se puede implementar como una secuencia constante de caracteres, por ejemplo “ARFS”.

Llegados a ese punto, el módulo de iniciador 220 comienza a ejecutar el archivo ejecutable del título, tal como se ilustra por medio de la etapa de procedimiento 526. En la realización ilustrativa, el título ejecutable se encuentra en forma de archivo ejecutable de Windows que está ubicado en el sistema de archivos que es implementado por el VxD de ARFSD 218 usando los datos que se recuperan por medio del VxD de RAFT 222.

El VxD de RAFT 222 comienza a recuperar los archivos y el directorio de título del servidor de RAFT 206, tal como se ilustra por medio de la etapa de procedimiento 528. Los bloques de datos que comprenden los directorios y los archivos de un título se recuperan del servidor de RAFT 206 usando el protocolo de RAFT y las instrucciones que se describen en el presente documento. En concreto, el VxD 222 recupera los bloques de datos del servidor de RAFT 206 de una forma con lectura anticipada y almacena en memoria caché los bloques de datos para facilitar un descifrado y una ejecución eficientes.

El VxD de ARFSD 218 utiliza el activador, en particular los datos de clave de descifrado, que se recibe del CAS 210 para descifrar los bloques de datos que se recuperan del servidor de RAFT 206 y para realizar una comprobación de integridad, tal como se ilustra en la etapa de procedimiento 530. Tal como se ha descrito en lo que antecede, el activador contiene una información criptográfica que es útil en el descifrado de los datos que están contenidos dentro del BRIQ antes de la ejecución del mismo. El VxD de ARFSD 218 mantiene una abstracción de instalación para el sistema operativo que crea la ilusión de que el sistema de archivos necesario para ejecutar el título está

instalado en el PC anfitrión local, tal como se ilustra por medio de la etapa de procedimiento 532. El proceso mediante el cual el VxD 218 mantiene las abstracciones de instalación que se describen con mayor detalle con referencia a la figura 6.

El testigo de RAFT que se recibe del CAS 210 incluye el campo de tiempo de fin tal como se describe con referencia a la figura 8 y su la descripción adjunta. Antes de la expiración del activador y el testigo de RAFT, el módulo de iniciador 220 emite una solicitud por medio de una conexión de RPC segura al servidor de CAS 206 para un par de activador / testigo de RAFT renovado, tal como se ilustra por medio de la etapa de decisión 534 y la etapa de proceso 536. El nuevo par de activador / testigo de RAFT se instala y se utiliza de una forma similar a la que se ha descrito previamente, tal como se ilustra por medio de la etapa de proceso 538.

Abstracción de instalación

De acuerdo con la presente invención, el título nunca está “instalado” realmente en el sistema anfitrión de cliente de SCDP. El soporte lógico de cliente de SCDP crea una abstracción de instalación, manteniendo la ilusión para el sistema operativo local de que el título que se está ejecutando en la actualidad está instalado en el ordenador anfitrión. Por lo tanto, cuando se ha terminado la ejecución del título, no queda evidencia alguna de que el título se ejecutara en el sistema de cliente anfitrión. No se deja fichero alguno que esté asociado con el título en la unidad de disco duro del sistema anfitrión, y no queda información alguna de estado de sistema operativo, por ejemplo, variables de registro que están asociadas con el título. El estado del sistema de cliente de SCDP después de que el título salga o el sistema se bloquee es el mismo que antes, excepto, posiblemente, por operaciones que son realizadas por otras aplicaciones, estado persistente, y cambios realizados por el usuario de la aplicación por ejemplo, datos o documentos guardados. La abstracción de instalación se logra con un procedimiento de carga del estado esperado de la aplicación, antes de la ejecución de la aplicación, de una forma tal que se puede deshacer la carga del estado cuando la aplicación sale sin afectar a los parámetros persistentes.

Cada BRIQ de acuerdo con la presente invención, y tal como se describe con referencia a la figura 12, incluye un sistema de archivos para uno o más títulos específicos. Tal como se describe en lo sucesivo en el presente documento, un autor de BRIQ utiliza un programa de utilidad de creación para extraer archivos seleccionados a partir de una aplicación y el directorio de instalación de la aplicación. El autor de BRIQ también extrae otra información tal como entradas de registro que pueden ser necesarias para la ejecución correcta de la aplicación. El programa de creación combina los archivos seleccionados y otra información y genera como una salida un sistema de archivos en forma de BRIQ, así como un conjunto de entradas de base de datos. El BRIQ se almacena en el servidor de RAFT. Las entradas de base de datos se almacenan en el servidor CAS y comprenden una información tal como información de generación de claves y valores de suma de comprobación de encabezado.

La figura 6 es un diagrama de flujo que ilustra las etapas de proceso que son realizadas por el cliente de SCDP 216 y los módulos 218-220 que están contenidos en el mismo para mantener la abstracción de instalación durante la ejecución de títulos de acuerdo con la presente invención. A continuación de la selección y la negociación para la compra de un título particular, el iniciador 220 y el VxD de ARFSD 218 montan el sistema de archivos, tal como se indica por medio de la etapa 600, y almacenan las entradas de registro asociadas en la unidad local del sistema anfitrión, tal como se indica por medio de la etapa 602. Una facilidad dentro del Gestor de Archivos de los sistemas operativos Windows 95, Windows 98 y Windows NT, así como la funcionalidad equivalente en el sistema operativo Unix, permite que el directorio de archivo y el contenido de un archivo ubicado de forma remota se “monte” o se acceda al mismo a través de una red informática, creando de ese modo una “unidad virtual” a partir de la cual se puede acceder a los datos. En la presente invención, montar el sistema de archivos comprende el uso de la técnica que se ha descrito en lo que antecede para acceder al servidor de RAFT a través de la interfaz de sistema operativo de cliente de SCDP. Montar el sistema de archivos puede dar como resultado el almacenamiento en memoria caché de la totalidad o de una porción de los bloques de datos procedentes de un BRIQ que contienen el contenido del título así como las entradas de registro que están asociadas con el título. La serie de entradas de registro se almacena de forma local en la memoria de sistema anfitrión del cliente de SCDP y puede incluir una información tal como el directorio en el que se han instalado los archivos del título, etc. El VxD de ARFSD 218 extrae adicionalmente las entradas de base de datos apropiadas a partir de la base de datos de CAS 212.

Usando la información de generación de claves a partir del activador, que se ha reenviado al cliente de SCDP mediante el servidor CAS, los bloques de datos procedentes del BRIQ se descifran y se ejecutan como un sistema de archivos de sistema operativo, tal como se indica por medio de la etapa 604. Los bloques de datos procedentes del BRIQ se almacenan en memoria caché de forma local en el cliente de SCDP según la necesidad por la totalidad de la ejecución de títulos. Durante la ejecución del programa, los controladores de dispositivo de sistema operativo, tales como los que están contenidos dentro de la porción de gestor de memoria virtual del sistema operativo, realizan solicitudes de entradas de registro que están almacenadas en la unidad física local. Tras la ejecución de la aplicación, el VxD de ARFSD 218 comienza a interceptar tales solicitudes operativas, tal como se indica por medio del bloque de decisión 606. Las llamadas, cuando sea aplicable, se satisfacen con entradas a partir de la lista de entradas de registro que están almacenadas de forma local, tal como se indica por medio de la etapa 608. Alguna información, no obstante, se escribe directamente en el sistema operativo usando las técnicas de escritura a través que se describen en lo sucesivo en el presente documento.

La intercepción de llamadas de sistema operativo y la satisfacción de estas solicitudes usando las entradas de registro almacenadas de forma local, continúa hasta la finalización de la ejecución de la aplicación, tal como se indica por medio de la etapa de decisión 610. Llegados a ese punto, el iniciador indica al VxD de ARFSD 218 que desmonte o desconecte el sistema de archivos, tal como se indica por medio de la etapa 612. Como resultado, las solicitudes de sistema operativo ya no se redirigen más a las entradas de registro almacenadas de forma local. Tanto las entradas de registro almacenadas de forma local como cualquier bloque de datos que se haya almacenado en memoria caché de forma local se pueden o bien suprimir o bien sobreescribir. Como resultado, el estado del cliente de SCDP antes de la ejecución del título o la aplicación se devuelve a su *statu quo* sin residuo alguno de instalación del título, excepto por cualquier dato de escritura a través limitado que el usuario desee retener de forma intencionada.

Almacenamiento local de escritura a través

Durante la generación de un BRIQ por el autor utilizando el programa de creación, los archivos y los directorios se pueden marcar con un atributo de “escritura a través”. Los BRIQ que contienen directorios o archivos de escritura a través pueden contener un contenedor con la etiqueta, LOCL. Este contenedor contiene la ruta completa de todos los directorios de escritura a través y la ruta de cualquier directorio, que no sea el directorio raíz, que contiene archivos de escritura a través, que se especifican con la etiqueta LDIR. Se permite que el usuario especifique que el nombre de ruta del directorio raíz para archivos almacenados de forma local. El nuevo nombre de ruta contiene el campo de Proveedor a partir del URN con el fin de asegurar la unicidad. Esta información se almacena en la etiqueta de RAÍZ en el contenedor de LOCL del título. Por defecto, el VxD de ARFSD notifica 0 bytes libres en la unidad local. Los BRIQ que no contienen directorio o archivo de escritura a través alguno siempre notificarán 0 bytes libres. La presencia de la etiqueta a en el contenedor de LOCL de un título especifica que el VxD de ARFSD debería notificar la cantidad de espacio libre en la unidad que contiene el directorio de almacenamiento local. Los títulos necesitan un contenedor de LOCL solo si es necesario que los mismos especifiquen unos valores no por defecto para la RAÍZ.

Cuando se carga un BRIQ que contiene directorios o archivos de escritura a través, es decir, que contiene un contenedor de LOCL en el encabezado, el iniciador dentro del cliente de SCDP crea un directorio para un almacenamiento local dentro del directorio de instalación de SCDP. Este directorio se obtiene del URN a menos que sea especificado un directorio por la etiqueta de RAÍZ en el contenedor de LOCL del título. El iniciador crea un subdirectorio en el directorio de almacenamiento local para cada directorio que se especifica con la etiqueta LDIR en el encabezado. Se pasa el nombre de ruta de raíz de la trayectoria de almacenamiento local así como si notificar el espacio libre en disco al VxD de ARFSD cuando se carga el BRIQ. Todos los archivos en áreas de almacenamiento local se suprimen cuando se desinstala el soporte lógico de iniciador y, opcionalmente, después de la salida del título. Estos archivos almacenados de forma local son persistentes por defecto. El iniciador ha de crear directorios en almacenamiento local para todos los directorios de escritura a través en un BRIQ.
Cuando se inicia un archivo de escritura a través, la información se toma del archivo en el área de almacenamiento local que tiene la misma convención de nomenclatura que los directorios que se han mencionado en lo que antecede. Si el archivo no existe en almacenamiento local, este se copia en primer lugar allí a partir del BRIQ. Puede que el archivo original en el BRIQ no esté comprimido o cifrado, aparte del cifrado de la totalidad del BRIQ. Cuando se abre un archivo de escritura a través, se abre la copia en un disco local, y todas las solicitudes en el manipulador de archivos de VxD de ARFSD se realizan en el manipulador de archivos real.

Servidor de Acceso Condicional (CAS, *Conditional Access Server*)

La figura 7A es un diagrama de bloques conceptual del Servidor de Acceso Condicional (CAS, *Conditional Access Server*) 700 y la base de datos asociada 750. En la realización ilustrativa, el CAS se puede implementar como una aplicación ejecutable en una plataforma compatible con POSIX.1 (Norma de IEEE 1003.1, 1998), tal como el sistema operativo Solaris® de Sun facilitado a nivel comercial por Sun Microsystems, Palo Alto, CA, o el sistema operativo Linux facilitado a nivel comercial por Red Hat Software, tales plataformas se pueden ejecutar en una arquitectura informática similar a la que se ilustra en la figura 1. La aplicación de CAS 702 comprende adicionalmente un módulo de interfaz de base de datos 704, una interfaz de llamadas a procedimiento remoto 706, un módulo de conversión de URN a URL 708, una fábrica de activadores 710 y un módulo de verificación de URL 712.

El módulo de interfaz de base de datos 704 interactúa con la base de datos de CAS 750 y se puede implementar usando productos de base de datos disponibles en el mercado. La base de datos 750 se puede usar para almacenar datos de permanencia a corto plazo, tales como los datos de permanencia de un testigo que solicita renovación, o datos de permanencia a largo plazo, tales como nombres de título, información de claves criptográficas, y otra información para títulos disponibles a través del sistema de SCDP. La base de datos 750 puede ser compartida por múltiples servidores de CAS 700, si se encuentra presente más de un servidor de CAS en una implementación a través de una red. La interfaz de base de datos 704 y la base de datos 750 comunican el lenguaje de consulta de base de datos según la norma de SQL. La norma de SQL es publicada por el Instituto Nacional Americano de Normas (ANSI, *American National Standards Institute*). La interfaz de base de datos 704 comprende un conjunto de objetos que filtran consultas que son recibidas por el servidor 700. Tales filtros son útiles a la hora de centrarse en o de personalizar el ámbito de una consulta de base de datos.

El servidor de CAS 700 está acoplado con el resto del sistema de SCDP por medio de la red 205, que en la realización ilustrativa es una red basada en protocolo de Internet que se implementa en forma o bien de una red de área local o bien de una red global. El servidor 700 interactúa con la red 205 a través de un módulo de llamadas a procedimiento remoto 706. El módulo 706 puede comprender código u objetos que siguen la norma de llamadas a procedimiento remoto de cómputo de redes abiertas, publicada por Sun Microsystems (RFC 1057 expedida por el grupo de tareas especiales de ingeniería en Internet). Tal RPC convencional define un código que controla el flujo y las llamadas a función entre dos entidades que intentan comunicarse de forma remota a través de una red. El módulo 706 se puede implementar con cualquier número de herramientas disponibles en el mercado que hacen que las llamadas a procedimiento remoto parezcan similares a las llamadas a función de subrutina. Un producto de este tipo, útil para la implementación del módulo 706, es la RPC Segura de Noblenet de Noblenet, Inc., Southborough, Massachusetts. La RPC Segura de Noblenet dota a una interfaz de RPC convencional de una capa de seguridad adicional.

El módulo de conversión de URN a URL 708 comprende código o una serie de objetos que, si se da un URN, consulta a la base de datos 750 y devuelven un URL correspondiente. Tales URN se reciben del módulo de iniciador del cliente de SCDP a través de la red 205. La base de datos 750 en la que se almacenan los URL se puede implementar como una base de datos secuencial que tiene una pluralidad de registros. El módulo 708 reenvía la consulta apropiada a la interfaz de base de datos 704 y recibe el URL apropiado a partir de la base de datos. El módulo 708 transmite entonces, a través del módulo de RPC 706, el URL correspondiente al cliente de SCDP a través de la red. Como alternativa, en un entorno en el que son utiliza un número limitado de títulos y / o servidores de contenido, los URL se pueden almacenar en un disco que está asociado con el servidor y el módulo 708 puede comprender una lógica de programa para llevar a cabo una conversión de tabla de consulta de un URN recibido.

El módulo de conversión 708 convierte las estructuras de datos de URN abstractas en unas estructuras de datos de URL específicas y se puede implementar con una serie de tablas de conversión y una lógica de comparación asociada. El módulo de verificación de URL 712 comprende código u objetos equivalentes que recibe una cadena de inicio a partir del servidor de comercio electrónico 202, tal como se explica con mayor detalle en lo sucesivo en el presente documento, coloca una marca de tiempo en la cadena de inicio y firma digitalmente la cadena de inicio a través del uso de un código de troceo y una clave de cifrado. En concreto, se puede adjuntar un código de autenticación de mensaje a la cadena de inicio según es recibido por el CAS 700. El código de autenticación de mensaje puede incluir un código de troceo que se genera de acuerdo con el algoritmo de troceo de MD5 e incluye adicionalmente una clave de cifrado que se puede generar de acuerdo con cualquier número de normas de cifrado incluyendo la Norma de Cifrado de Datos (DES, *Data Encryption Standard*). La cadena de inicio firmada digitalmente se reenvía entonces al servidor de comercio electrónico 202 para su transmisión de vuelta al navegador web del sistema anfitrión de cliente tal como se describe en el presente documento.

En el sistema de SCDP, el activador sirve como un mecanismo para distribuir una información de generación de claves a procesos de cliente potencialmente inseguros. El módulo de generación de activadores 710 del servidor 700 comprende código u objetos apropiados que generan una serie de códigos de bytes y adjunta una clave criptográfica a la serie de códigos de bytes, recuperándose la clave, en una implementación, de la base de datos 750. La implementación del módulo de generación de activadores depende en parte de la sofisticación de los activadores que se utilizan dentro del sistema de SCDP. Para un activador que comprende una serie de códigos de bytes y una clave que está adjunta al mismo o integrada en el mismo, el módulo de generación de activadores 710 tiene la implementación que se ha descrito en lo que antecede. En realizaciones alternativas, en las que la clave está integrada en el activador de una forma más segura, por ejemplo, plegando la clave para dar la secuencia de códigos de bytes, se requeriría una lógica y / u objetos adicionales para implementar tales funciones dentro del módulo 710. Por ejemplo, en lugar de adjuntar una clave a una serie de códigos de bytes, se puede insertar en el activador una secuencia de códigos de bytes que realizan una función, tal como la generación de un número o la realización de otras operaciones lógicas. En una realización de este tipo, el módulo 710 puede incluir una lógica para seleccionar de forma aleatoria de entre una de un número de secuencia de códigos de bytes o técnicas que se describen en el presente documento como técnicas de ofuscación de código. Con una realización de este tipo, el módulo 710 es capaz de generar, de forma aleatoria, activadores con un grado más alto de seguridad. Como alternativa, con unas implementaciones de activador más sofisticadas, el módulo de activador 710 puede generar un activador a través de una llamada a una rutina de generación de activadores que puede residir en una biblioteca externa.

Los módulos de CAS que se han descrito en lo que antecede pueden realizar cinco funciones primarias dentro del sistema de SCDP. En primer lugar, el CAS dota a los usuarios de los activadores criptográficos que permiten un uso de única vez de un contenido de BRIQ cifrado. En segundo lugar, el CAS asegura que el sistema de SCDP puede realizar con precisión un seguimiento del uso de títulos y soportar un modelo de seguridad en el que resulta muy difícil el desarrollo de un cliente “pirateado” diseñado para robar el uso. En tercer lugar, el CAS proporciona unos testigos de autorización de RAFT con tiempo de vida limitado que están firmados con una clave privada de CAS y agrupados con un activador. El cliente de RAFT incluye el testigo de autorización con sus solicitudes de RAFT. El servidor de RAFT usa el testigo para verificar el derecho de un cliente a acceder al contenido solicitado. En cuarto lugar, el CAS interactúa con el sistema de facturación de soporte lógico de comercio electrónico para “resolver” transacciones. La resolución de la transacción no se realiza durante la negociación de compra sino que se retrasa hasta que el CAS está seguro de que el usuario final ha sido capaz de ejecutar el contenido con éxito. La

compleción de la primera renovación de activador es una indicación de que el título se está ejecutando con éxito. En quinto lugar, el CAS mantiene una base de datos para la notificación de uso de títulos y el seguimiento de activadores.

Se pueden asociar tres tipos de registros con el CAS. En primer lugar, la actividad de CAS está registrada con un registro de texto UNIX convencional. Este registro solo tiene por objeto fines de diagnóstico. En segundo lugar, el CAS registra transacciones en la tabla de base de datos de CAS, para fines de notificación y para el seguimiento de activadores. Estos registros son además de los que son mantenidos por el sistema de comercio electrónico, que se usan para fines de facturación real. En tercer lugar, la propia base de datos de CAS mantiene registros de transacción internos, que son los mecanismos que se usan para asegurar que las transacciones de base de datos se han completado o revertido con éxito. Tal funcionalidad puede ser interna a la base de datos de CAS. En la realización ilustrativa, el CAS usará un soporte lógico de mantenimiento de base de datos disponible en el mercado tal como el facilitado a nivel comercial por Oracle Software para asegurar que una compra se ha confirmado o revertido. Las transacciones de base de datos son diferentes de las transacciones financieras que se han descrito en lo que antecede. Una transacción financiera puede ser una transacción de base de datos, pero muchas otras transacciones tales como la actualización de un nombre de usuario, pueden ser transacciones de base de datos.

En la realización ilustrativa, el CAS soporta una interfaz de administración con la que un administrador de sistemas puede supervisar el estatus de CAS, por ejemplo, el número actual de subprocesos de conexión de base de datos en uso y el número actual de conexiones de usuario, es decir, los subprocesos de conexión en uso. Además, se puede facilitar una información estadística tal como el número máximo de conexiones de usuario y de base de datos que se usan desde el arranque; el número de veces desde el arranque que las conexiones de base de datos o usuario han alcanzado un límite previamente determinado.

El cliente de SCDP interactúa con el CAS por medio de una biblioteca de cliente. La biblioteca de cliente puede ser específica de cada plataforma de cliente, debido a que esta usa procedimientos nativos de la plataforma para comunicarse con la GUI de cliente de SCDP. En la realización ilustrativa, es decir, la plataforma de Win32, la biblioteca de cliente se denomina CASLIB32. La biblioteca de cliente exporta las clases de interfaz de CAS CCAS, que representa el transporte al CAS, y CCasSession, que representa una sesión de cliente específica. Una Interfaz de Programación de Aplicaciones (API, *Application Program Interface*) permite que el cliente de CASLIB32 negocie para múltiples títulos de forma simultánea mediante el uso de múltiples sesiones. La API también exporta clases adicionales que representan una información pasada a y recibida del CAS que aíslan la interfaz de CAS de los detalles concretos del protocolo de transporte. Tales procedimientos se pueden implementar como CActivator, CUrl, etc. La CCAS responde de forma asíncrona al cliente mediante el envío de mensajes de Windows.

Soporte de CAS para activadores

La implementación más simple de un activador es una rutina de códigos de bytes que tiene la clave para un BRIQ dado que está compilado en el activador. Con esta implementación de activador, el CAS autentica el cliente, identifica el BRIQ comprado, construye el código de bytes de activador y descarga el activador. El cliente de SCDP puede cerrar entonces la conexión y ejecutar el título. Tales activadores se pueden generar con antelación y recuperarse directamente de una base de datos mediante la fábrica de activadores de CAS 710 y la interfaz 704.

En una implementación de activador más sofisticada, el activador es consciente de un algoritmo criptográfico, y solicita una clave a partir del CAS. El CAS tiene información de autenticación y datos de seguridad a partir del flujo existente, y puede tener una respuesta de RPC previamente definida para “solicitar clave” con cualquier argumento que sea necesario.

En otra implementación, el activador puede tener un código arbitrario para algún nuevo mecanismo, requiriendo posiblemente múltiples fases. El activador puede realizar una llamada a procedimiento remoto para el CAS con argumentos opacos y una especificación de una “técnica”, tal como se explica en lo sucesivo en el presente documento. El CAS distribuye entonces los datos opacos a la técnica, que devuelve datos opacos al cliente o realiza otras llamadas, o llama a otros servicios. Si el CAS tiene su propio intérprete, el CAS puede recuperar el código para el activador y la técnica a partir de la base de datos. Si todos los activadores se generan previamente, puede haber muchos activadores posibles para cualquier técnica singular. Como alternativa, una base de datos de ofuscaciones y un conjunto de reglas para cómo combinar las mismas pueden ser mantenidos por el CAS.

El CAS selecciona un activador apropiado para un cliente, un producto y una compra dados. El CAS entrega el activador, y “soporta” el activador a través de RPC adicionales. Muchas RPC de CAS pueden estar previamente definidas, tal como una “solicitar clave” simple para un BRIQ dado. Tales RPC se pueden restringir basándose en el activador particular seleccionado. Por ejemplo, a la mayor parte de los clientes no se les permitirá la llamada de “solicitar clave” simple, pero se les requeriría que realizaran cualquier llamada cuyo uso por parte del activador espere la técnica.

Haciendo referencia a la figura 7B, se ilustra un diagrama de flujo que ilustra las etapas de proceso que son realizadas por el CAS 700 durante el proceso de abono y de ejecución de títulos. En concreto, el CAS 700 recibe una cadena de inicio, tal como se describe con referencia a la figura 9 y su explicación adjunta, a partir del servidor

de comercio electrónico, tal como se ilustra en la etapa 720. A continuación, el CAS “firma” digitalmente la cadena de inicio, tal como se indica en la etapa de procedimiento 722. El CAS “firma” la cadena de inicio con una clave criptográfica privada. La cadena de inicio firmada se reenvía entonces a partir del CAS 700 al cliente de SCDP que se está ejecutando en un sistema anfitrión que está conectado con la red de banda ancha, tal como se ilustra por medio de la etapa de proceso 724. El cliente de SCDP extrae el URN a partir de la cadena de inicio, tal como se describe en el presente documento con referencia a las figuras 5A-C y sus descripciones adjuntas, y transmite el URN al CAS 700. El CAS 700 recibe el URN a partir del cliente de SCDP, tal como se ilustra por medio de la etapa 726, y realiza una conversión del URN a un URL, tal como se ilustra por medio de la etapa de procedimiento 728. Tal como se ha descrito en lo que antecede, el CAS 700 realiza la conversión de URN a URL usando el módulo 708 tal como se ha descrito en lo que antecede. Tal conversión puede incluir una consulta de la base de datos 750 o el uso de un algoritmo de consulta en tabla, dependiendo de la implementación del módulo 708. El CAS 700 transmite la lista de URL al cliente de SCDP, que también se ilustra en la etapa de procedimiento 728. A continuación, el CAS 700 recibe una solicitud de autorización de compra a partir del cliente de SCDP, tal como se ilustra por medio de la etapa de procedimiento 730. La solicitud de autorización de compra a partir del cliente de SCDP incluye la cadena de inicio. El CAS 700 verifica entonces la cadena de inicio para determinar si la cadena de inicio había sido firmada previamente por este o, en una implementación con múltiples servidores de acceso condicional, por otro servidor de CAS autorizado, tal como se ilustra por medio de la etapa de procedimiento 732. El CAS 700 genera entonces un activador para el cliente que solicita una autorización de compra, tal como se ilustra por medio de la etapa de proceso 734. La generación de activadores tiene lugar de acuerdo con la implementación específica del módulo 710 del servidor CAS, tal como se describe en el presente documento.

A continuación, el CAS 700 transmite el activador así como un testigo de RAFT al cliente de SCDP, tal como se ilustra en la etapa de procedimiento 736. El CAS 700 recupera el testigo de RAFT de la base de datos 750. El testigo de RAFT tiene el formato que se ilustra en la figura 8 y tal como se describe en las porciones relevantes en el presente documento. El activador y el testigo de RAFT posibilitan que el cliente de SCDP acceda al título deseado y que comience la ejecución de los datos de título tal como se describe en el presente documento. Llegados a este punto, el CAS no emprenderá acción adicional alguna en relación con el cliente de SCDP específico hasta que se recibe una solicitud de renovación de testigo de activador a partir del cliente de SCDP, tal como se ilustra por medio de la etapa de decisión 738. Tras la recepción de la primera solicitud de renovación a partir del cliente de SCDP, el CAS 700 publica la compra del título en el servidor de comercio electrónico, tal como se ilustra por medio de la etapa de procedimiento 740. La publicación de la transacción con el servidor de comercio electrónico comprende la confirmación registrada real de que el usuario ha pagado por el título identificado. Tal publicación se retarda hasta la primera solicitud de renovación para asegurar que el título se está ejecutando de forma apropiada en el cliente de SCDP. El mecanismo de tiempo de espera dentro del activador que se envía inicialmente desde el CAS al cliente de SCDP expira después de un intervalo predeterminado, indicando que el título se está ejecutando de forma apropiada. El CAS emite un nuevo testigo, tal como se ilustra en la etapa de procedimiento 742, y transmite el par al cliente de activador de SCDP solicitante. El tiempo de vida de testigo de RAFT, tal como se ilustra por medio de los campos de tiempo de inicio y de tiempo de detención del testigo de RAFT, puede ser más prolongado que el tiempo de vida del testigo que se transmite inicialmente al cliente de SCDP con el activador. Las solicitudes posteriores de renovación de activador/testigo a partir del cliente de SCDP no darán lugar a que el CAS publique la compra del título en el servidor de comercio electrónico. Tal como se ha descrito en lo que antecede, toda la comunicación entre el cliente de SCDP y el CAS tiene lugar a través de una conexión de RPC segura, tal conexión se puede establecer usando un producto disponible en el mercado que sigue la norma de RPC.

Tal como puede ser apreciado por los expertos en la materia, el proceso que se bosqueja en la figura 7D resalta las etapas que se ejecutan mediante el CAS en relación con un cliente de SCDP particular que terminará cuando finaliza la ejecución de títulos. Será obvio a los razonablemente expertos en la materia que el CAS se puede implementar como una aplicación multitarea en la que diversos subprocesos separados están ejecutando en la actualidad diversas etapas del proceso ilustrado. Por consiguiente, al tiempo que se atienden las solicitudes de un cliente de SCDP específico, el CAS también puede estar atendiendo de forma concurrente las solicitudes a partir de otros clientes de SCDP.

Transporte de RPC

El CAS y la CASLIB32 se comunican a través de una biblioteca de llamadas a procedimiento remoto basada en normas, tal como la RPC Segura de Noblenet. El cliente de SCDP realiza llamadas síncronas al CAS, que asigna las mismas a un subproceso para su procesamiento. La CASLIB32 presenta una interfaz asíncrona a su GUI adjunta, por lo tanto esta pone en cola, de forma interna, las solicitudes de RPC síncronas y las coloca a partir de un subproceso en segundo plano. Para proporcionar un alto caudal de transacciones, el CAS mantiene un grupo de subprocesos listos que se pueden usar para ejecutar tareas. El grupo de subprocesos es una clase de C++ reutilizable. Las tareas entrantes se interceptan en la capa de RPC, se ponen en cola en el grupo de subprocesos y, con el tiempo, se procesan en un subproceso en contraposición a procesarse en línea. El grupo de subprocesos permite que el CAS procese unas tasas de transacción simultánea más altas y que presente un mejor desempeño en caso de picos de carga cortos. Es necesario que las llamadas RPC atribuyan una memoria segura para subprocesos que se pueda marcar y liberar posteriormente, debido a que las memorias intermedias no se pueden liberar hasta que el transporte de RPC se realiza enviando las mismas. El CAS usa una clase de grupo de memoria de C++ reutilizable que puede suprimir la memoria por id de subproceso.

El CAS se puede implementar como un servidor sin estados, como un servidor web. Un servidor sin estados tiene la ventaja de que este se puede ajustar a escala con facilidad mediante la distribución de más máquinas de servidor y el uso de un soporte lógico “por orden cíclico” para distribuir las conexiones entrantes a los servidores, debido a que no es necesario que las solicitudes posteriores de un cliente de SCDP vayan al servidor con el que este las conectó originalmente. El CAS mantiene un flujo de TCP de zócalo conectado entre solicitudes, se podría adjuntar por lo tanto alguna información, tal como una clave de sesión de transporte. Si se abandona esta conexión, la CASLIB32 intentará volver a conectar, potencialmente con un proceso de CAS diferente, por lo tanto es preferible la inserción del estado en la CASLIB32 o en la base de datos.

Para facilitar un alto volumen de transacciones, el CAS está diseñado para hacer uso de un grupo de múltiples conexiones de base de datos activas. Los subprocesos de servidor solicitan conexiones del grupo, que reconecta conexiones inactivas en el segundo plano según sea necesario para reducir al mínimo la latencia de conexión de base de datos. El grupo de conexión de base de datos se implementa como una clase de C++ reutilizable. El CAS usa una interfaz de base de datos abstracta que se denomina DBObject, que se implementa como una clase de C++ reutilizable y permite que el CAS se porte fácilmente a otras bases de datos.

Testigo de RAFT

Para mejorar el modelo de seguridad global del sistema de SCDP, el CAS dota al cliente de SDCP de un testigo de autorización de RAFT firmado. El testigo de RAFT autoriza el acceso de un cliente de SCDP particular a un URN particular, durante un periodo de tiempo especificado. El CAS firma digitalmente el testigo de RAFT, usando algoritmos de firma digital de clave pública normalizados. Con el fin de acceder a un contenido ejecutable en un servidor de RAFT, el VxD de RAFT ha de presentar su testigo a ese servidor. El servidor de RAFT verifica la firma digital del CAS y, entonces, verifica los contenidos del testigo. El testigo de RAFT 800 es válido para cualquier número de los servidores de RAFT dentro del dominio administrativo de un CAS; es decir, un proveedor de servicios de banda ancha puede instalar múltiples servidores de RAFT en su red, y el testigo de RAFT podría ser admitido por cualquiera de los mismos.

En la realización ilustrativa, el testigo de RAFT se implementa como una estructura de datos que tiene el formato que se ilustra en la figura 8. El testigo de RAFT 800 comprende un URN, una longitud de URN 804, un tiempo de inicio 806, un tiempo de fin 808, una dirección de IP 810 y una firma de CAS 812. El URN 802 y su longitud asociada 804, definen el título específico que desbloqueará el testigo de RAFT. El tiempo de inicio 806 y el tiempo de fin 808 definen el tiempo de vida del testigo. El formato del URN descrito se ha descrito previamente. El testigo de autorización de RAFT contiene la dirección de IP del cliente de RAFT como un valor de 32 bits en orden de bytes de red, el URN solicitado, y tiempos de inicio y de expiración de 32 bits. Los tiempos se definen como “segundos desde la Época” de POSIX 1003.1-1988 o aproximadamente segundos desde 00:00:00 GMT, 1 de enero de 1970. El CAS firma el testigo con la clave privada del grupo de CAS de tal modo que el servidor de RAFT puede validar su autenticidad. El servidor de RAFT denegará el acceso si el tiempo actual del servidor no se encuentra dentro de la ventana del testigo. La dirección de IP define la dirección de red del cliente de SCDP que solicita el activador/testigo. El servidor de RAFT denegará el acceso si el cliente de SCDP que proporciona el testigo no tiene la misma dirección de IP, evitando de ese modo que otro cliente use un testigo robado.

El testigo de RAFT se transfiere al cliente como parte del activador. Los testigos de RAFT se renuevan junto con los activadores. El activador se construye con un mecanismo de tiempo de vida. El cliente de SCDP emite una solicitud de CAS, por medio del mecanismo de RPC, para renovar la combinación de activador/testigo antes de la expiración del activador existente.

Servidor y protocolo de transporte de archivos de acceso aleatorio

La figura 10 ilustra de forma conceptual un diagrama de bloques del servidor de RAFT 1000 y su base de datos 1050 adjunta. En la realización ilustrativa, el servidor de RAFT 1000 se puede implementar como una aplicación ejecutable en una plataforma compatible con POSIX.1 (Norma de IEEE 1003.1, 1998), tal como el sistema operativo Solaris® de Sun facilitado a nivel comercial por Sun Microsystems, Palo Alto, CA, o el sistema operativo Linux facilitado a nivel comercial por Red Hat Software, tales plataformas se pueden ejecutar en una arquitectura informática similar a la que se ilustra en la figura 1.

El servidor de RAFT se puede implementar como una aplicación de RAFT 1002 y un agente de maestro de Protocolo Simple de Gestión de Redes (SNMP, *Simple Network Management Protocol*) 1004 que se está ejecutando encima del sistema operativo. Un producto disponible en el mercado conveniente para implementar el agente de maestro de SNMP 1004 es el producto Emanate facilitado a nivel comercial por SNMP Research, Inc. El agente de maestro 1004 se comunica con la red 205 usando interfaces de programación de aplicaciones publicadas de acuerdo con las normas de SNMP.

La aplicación de RAFT 1002 comprende un módulo de entrada/salida de archivos de POSIX (*Portable Operating System Interface Standard*, Norma de Interfaz de Sistema Operativo Portátil) 1006, una la interfaz de sistema de archivos 1008 y un módulo de instrumentación de SNMP 1010 (es decir, el subagente de SNMP de RAFT) y un módulo de interfaz de protocolo de red/RPC/RAFT 1012.

El módulo de instrumentación de SNMP 1010 contiene objetos o código correspondiente que recopila una información estadística y logística útil para un administrador de sistemas a la hora de limitar el ancho de banda de la red para mejorar el desempeño de la red. En ese sentido, el módulo 1010 es un elemento opcional del servidor de RAFT 1000.

El módulo de Protocolo de RAFT de RPC 1012 interactúa con la red basada en IP 205 usando un protocolo de RPC patentado tal como se define en el presente documento. El módulo 1012 incluye el código y / o los objetos necesarios para implementar el protocolo y para verificar los contenidos del testigo de RAFT.

El módulo de entrada/salida de archivos 1006 puede ser una implementación orientada a objetos de acuerdo con la norma de POSIX 1003.1 publicada por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, *Institute of Electrical and Electronic Engineers*). El módulo de E/S de POSIX 1006 proporciona una abstracción de interfaz de sistema de archivos local para los discos de memoria 1050. La memoria 1050, que se ilustra esquemáticamente de forma conceptual en la figura 10, se usa para almacenar múltiples títulos en las formas de los BRIQ. En la realización contemplada, la porción de encabezado de un BRIQ que está sin codificar y la porción de cuerpo de un BRIQ, que está codificada, se almacenan conjuntamente. No obstante, se accede a los mismos de forma independiente entre sí utilizando el módulo 1006 y 1008. El módulo de interfaz de sistema de archivos 1008 contiene una lógica de programa que recibe solicitudes de un BRIQ particular y pone en correspondencia el BRIQ con el directorio y el archivo en el que se almacena el mismo en la memoria 1050. De esta forma, la interfaz de sistema de archivos 1008 funciona como una interfaz entre la solicitud de red a partir del sistema de SCDP y la memoria 1050. En la realización ilustrativa, la memoria 1050 se puede implementar como uno o más discos, por ejemplo, un conjunto de discos de tipo RAID o una granja de discos. El módulo de interfaz de sistema de archivos 1008 interactúa con el módulo de entrada/salida de archivos 1006 y el módulo de protocolo de red 1012 e implementa una lógica de programa para acceder a archivos y BRIQ tal como se describe en el presente documento.

El agente de maestro de SNMP 1004 proporciona servicios de protocolo de SNMP en nombre del subagente de SNMP de RAFT, que está incrustado dentro de la aplicación de RAFT. La aplicación de RAFT usa su subagente de SNMP para hacer que su gestión sea accesible a un gestor de SNMP remoto.

Las siguientes etapas describen la interacción entre el cliente de SCDP y el servidor de RAFT, el iniciador inicia un título. El iniciador entra en contacto con el servidor CAS para obtener una lista de los URL que se corresponden con el URN solicitado. Un URL identifica la ubicación de un BRIQ particular, incluyendo el servidor de RAFT en el que reside este. Para cada URL de RAFT, se puede devolver un peso para ayudar a seleccionar el URL más apropiado. Un URL es más deseable cuando este tiene un valor de peso más alto.

A continuación de la conversión de URN a URL por el CAS, el cliente de SCDP envía al CAS una solicitud de compra que se ha descrito previamente en el análisis de los intercambios de CAS. En respuesta a la solicitud de compra, el servidor CAS dota al cliente de SCDP de un activador que contiene el testigo de autorización de RAFT para el URN seleccionado. Obsérvese que el testigo de autorización es válido para cualquiera de los URL que están asociados con el URN seleccionado.

El iniciador examina entonces la lista de URL para determinar si se encuentra presente cualquier URL de RAFT. Si se encuentran presentes URL de RAFT, el iniciador envía solo la lista de URL de RAFT junto con el testigo de acceso de RAFT a un VxD de ARFSD que reenviará esta información al cliente de RAFT, es decir, el VxD de RAFT del cliente de SCDP. El iniciador también proporciona un peso para cada uno de los URL de RAFT. Estos pesos pueden ser diferentes de los proporcionados por el CAS durante la conversión de URN a URL. El cliente de RAFT establece entonces una conexión con uno de los servidores de RAFT que son especificados por la lista de URL. El cliente de RAFT puede contener la lógica de programa apropiada que posibilita que este use los pesos provistos con los URL para decidir con qué servidor de RAFT establecer contacto en primer lugar.

El cliente de RAFT intenta entonces abrir un BRIQ en el servidor de RAFT 1000. El cliente especifica una versión de protocolo, el nombre de ruta (a partir del URL) y el testigo de acceso de RAFT. La versión de protocolo es un valor de 32 bits que se usa para verificar que el cliente de RAFT y el servidor de RAFT son compatibles con protocolo. Para validar el acceso, el servidor de RAFT verifica que el URN que se proporciona en el testigo es uno de los enumerados en el encabezado de BRIQ. El servidor de RAFT 1000 comprueba los tiempos de inicio y de expiración del testigo de RAFT durante la apertura. Si la RAFT_OPEN tiene éxito, el servidor de RAFT devuelve un manipulador de archivos de RAFT y un ID único para el BRIQ, por ejemplo un valor de troceo del marcador de BRIQ, que se usa para el almacenamiento en memoria caché.

Con el fin de que el servidor de RAFT valide el tiempo de expiración, el tiempo de servidor de RAFT está sincronizado con el CAS dentro de un intervalo predeterminado. Por lo tanto, el servidor de RAFT acepta unos tiempos de inicio antes del tiempo actual y no deniega el acceso hasta después de la expiración del intervalo. Se propone que el tiempo de expiración del testigo sea algún múltiplo del tiempo de conexión persistente de Activador, más un tiempo adicional para manejar las latencias variables de red y de servidor.

En cada solicitud de lectura de RAFT posterior, el servidor de RAFT comprueba que el testigo de acceso no ha expirado. El servidor de RAFT dará un fallo al realizar cualquier solicitud que tenga lugar cuando el servidor no tenga

un testigo de acceso válido para ese cliente particular.

Con el tiempo, el testigo de acceso de RAFT expirará. El mecanismo de conexión persistente de activador del cliente de SCDP es responsable de obtener un nuevo testigo de RAFT antes de que expire el testigo actual. Esto asegura que los testigos de RAFT se renuevan de una forma oportuna de tal modo que no tendrán lugar fallos de acceso en unas condiciones operativas normales. Cuando el cliente de RAFT envía el testigo al servidor de RAFT durante una RAFT_OPEN, el cliente de RAFT ha de computar durante cuánto tiempo es válido el testigo a partir del tiempo de inicio y de expiración. Debido a que el cliente de RAFT no puede verificar la legitimidad de los contenidos de testigo sin la clave pública del CAS, el cliente de RAFT ha de aguardar una RAFT_OPEN con éxito para determinar que el testigo es válido antes de establecer su tiempo de renovación. No obstante, el tiempo de renovación se basa en cuándo recibió el testigo el cliente de RAFT y no en cuándo se completó la RAFT_OPEN. Para asegurar un acceso ininterrumpido al servidor, el cliente de RAFT solicita un nuevo testigo de acceso de RAFT a partir del CAS antes de que expire el testigo. Tras la recepción del nuevo testigo de acceso de RAFT, el cliente de RAFT enviará una operación de RAFT_REFRESH con el testigo recién obtenido al servidor de RAFT.

Cuando el cliente de RAFT ha acabado de acceder a un BRIQ, el cliente de RAFT envía un mensaje de RAFT_CLOSE con el manipulador de archivos de RAFT. Si el servidor de RAFT pierde la conexión con el cliente de RAFT, todos los archivos abiertos que se corresponden con esa conexión se cierran de forma automática.

Definición de encabezado de paquete de RAFT

Todas las comunicaciones de acuerdo con el protocolo de RAFT contienen un encabezado de paquete de RAFT 1100, tal como se ilustra en la figura 11. El encabezado de paquete de RAFT 1100 se puede implementar como una estructura de datos que comprende un campo de datos de número de procedimiento 1102, un campo de datos de número de secuencia 1104, un campo de datos de longitud de paquete 1106, y un campo de datos de estatus 1108. El campo de número de procedimiento 1102 indica el tipo de mensaje de protocolo de RAFT y se puede implementar en forma de número entero. El campo de número de secuencia 1104 se usa para hacer coincidir solicitudes con respuestas y se puede implementar en forma de número entero. El número de secuencia solo es único por conexión. El campo de longitud de paquete 1106 indica el tamaño de los datos de paquete, sin incluir el tamaño del encabezado, y se puede implementar en forma de número entero. El campo de estatus 1108 indica el estatus a partir de la solicitud de RAFT y se puede implementar en forma de número entero. Un estatus distinto de cero indica que falló la solicitud. Diferentes mensajes de protocolo devolverán diferentes códigos de estatus. No obstante, un estatus de cero indica que la solicitud se completó con éxito. Un estatus distinto de cero da como resultado que el campo de longitud se ajuste a cero, indicando que no se devolverá dato de paquete alguno si falla una solicitud. De acuerdo con el protocolo de RAFT, el encabezado de paquete va seguido de los datos de paquete de RAFT.

Mensajes de Protocolo de RAFT

El protocolo de RAFT consiste en cuatro mensajes de protocolo diferenciados que posibilitan el acceso a los BRIQ y la gestión de los testigos de RAFT. Después del establecimiento de la conexión de TCP, el mensaje de protocolo de RAFT inicial contiene la versión de protocolo como uno de sus argumentos con el fin de identificar la versión de protocolo del solicitante. Una lista y una descripción de los mensajes de protocolo de RAFT tal como sigue. La función de RAFT_OPEN es llamada con una versión de protocolo, una longitud de testigo, un testigo de acceso de RAFT, una longitud de ruta y un nombre de ruta completo terminado en nulo. Tras el éxito, el resultado es un manipulador de archivos de RAFT, un ID de RAFT, y la longitud de lectura máxima que es soportada por el servidor de RAFT. El ID de RAFT se puede usar para generar un marcador de memoria caché de cliente de SCDP. El ID de RAFT puede ser el ID de BRIQ para posibilitar un almacenamiento en memoria caché consistente a través de múltiples servidores de RAFT en el caso de que tenga lugar una conmutación por error. La longitud de lectura máxima tiene por objeto informar al cliente de RAFT acerca de cuántos datos puede solicitar el mismo durante una operación de RAFT_READ.

La función de RAFT_REFRESH_TOKEN posibilita que el cliente de RAFT actualice el servidor de RAFT con un testigo de acceso de RAFT más nuevo y es llamada con una longitud de testigo, un testigo de acceso de RAFT y un manipulador de archivos de RAFT. Tras el éxito, el nuevo testigo de acceso de RAFT sustituye el testigo actual que está asociado con el manejador especificado, aumentando de forma efectiva el tiempo de expiración del testigo. El testigo actual se conservará si el nuevo testigo no es válido. Esta función no devuelve dato alguno, pero el estatus en el encabezado se actualiza para reflejar éxito o fallo.

La función de RAFT_READ es llamada con el manipulador de archivos de RAFT que se devuelve de la llamada de RAFT_OPEN, un desplazamiento de 64 bits y una longitud. El manipulador de archivos de RAFT se ha de asociar con un testigo de acceso válido con el fin de acceder a los datos solicitados.

La función de RAFT_CLOSE se usa para cerrar un manipulador de archivos de RAFT abierto. La llamada toma un manipulador de archivos de RAFT y no devuelve dato alguno. No obstante, el estatus en el encabezado se actualiza para indicar éxito o fallo.

Cadena de inicio

La figura 9 ilustra una cadena de inicio 900.

La cadena de inicio 900 se puede implementar como una estructura de datos que comprende un campo de datos de URN 902, un campo de datos de ID de Tienda 904, un campo de datos de tipo de mercancía 906, un campo de datos de dominio de abono 908 y un campo de datos de cantidad 910, tal como se ilustra en la figura 9. El URN 902 que identifica de forma única el contenido deseado y se puede implementar, tal como se describe en el presente documento. El ID de Tienda 904 identifica un escaparate electrónico específico para el sistema de comercio electrónico y se puede implementar en forma de cadena de caracteres numéricos o alfanuméricos o un número entero. Los ID de Tienda se usan para separar las transacciones de diferentes escaparates electrónicos para notificar una compra. Múltiples escaparates electrónicos pueden compartir un ID de Tienda si estos están representando realmente a la misma organización. El tipo de mercancía 906 indica si la transacción debería ser una compra a través de un abono o a través de una microtransacción y se puede implementar en forma de cadena de caracteres numéricos o alfanuméricos o un número entero. Una transacción de abono es un único pago para un uso ilimitado de un título o conjunto de títulos a lo largo de un periodo de tiempo especificado. Una microtransacción es un cargo contra una cuenta de débito de usuario, y se usa para soportar el modelo de pago de “pago por uso único”. El dominio de abono 908 indica si la transacción está cubierta por una oferta de abono específica, por ejemplo, “Paquete Semanal de Juegos de Moda” o “Paquete de Aplicaciones para Oficinas Pequeñas”, aplicable a la compra. El dominio de abono se puede implementar en forma de cadena de caracteres numéricos o alfanuméricos o un número entero. El campo de cantidad 910 indica la cantidad de compra de la microtransacción y se puede implementar en forma de número entero.

Los contenidos de la cadena de inicio 900 son generados por el módulo de extremo frontal de servidor de comercio electrónico 1408 tal como se ilustra en la figura 14. El CAS firma digitalmente la cadena de inicio 900, usando, por ejemplo, un algoritmo de firma digital de clave pública normalizado. A continuación de lo anterior, la cadena de inicio 900 comprende un campo de firma de CAS adicional 912 que identifica la clave privada del grupo de CAS. La cadena de inicio se envía al cliente de SDCP por medio del sistema de comercio electrónico, como parte del proceso de cumplimiento. El cliente de SDCP pasa la cadena de inicio de vuelta al CAS durante sus negociaciones previas al inicio con el CAS, tal como se explica en el presente documento.

Sistema de comercio electrónico

Una aplicación de soporte lógico de comercio electrónico, al que se hace referencia en lo sucesivo en el presente documento como sistema de comercio electrónico, conveniente para su uso es Transact 4.0, facilitado a nivel comercial por Open Market, Cambridge, MA. El soporte lógico de comercio electrónico se usa para gestionar cuentas de usuario y llevar a cabo transacciones financieras, incluyendo 1) mantener información de cuentas de usuario, 2) gestionar la compra y el pago, 3) recopilar y verificar información de tarjetas de crédito, y 4) resolver transacciones.

Haciendo referencia de nuevo a la figura 2, el servidor de comercio electrónico 202 comprende una aplicación de servidor que se está ejecutando en una arquitectura informática similar a la que se describe con referencia a la figura 1. La aplicación se puede diseñar para operar sobre un sistema operativo tal como el sistema operativo Solaris de Sun u otros sistemas operativos diseñados para ejecutar aplicaciones de tipo servidor. Haciendo referencia a la figura 14, el servidor de comercio electrónico 14 comprende una plataforma de soporte físico 1402 en la que se ejecuta un sistema operativo 1404. La aplicación de servidor de comercio electrónico 1406 real presenta un módulo de extremo frontal 1408 y un módulo de extremo posterior 1410 a los diversos otros componentes del sistema de SCDP 200. En concreto, el módulo de extremo frontal 1408 del servidor 1400 se puede implementar para producir un extremo frontal de servidor de web para los otros componentes del sistema de SCDP 200 a través de la red 205. Un extremo frontal de este tipo es similar a otros servidores web que existen en la actualidad en Internet. El módulo del extremo posterior 410 del servidor 1400 interactúa con la base de datos de facturación 204 e implementa la lógica y / o los objetos necesarios para consultar la base de datos y ejecutar transacciones y microtransacciones que están asociadas con la negociación y la compra de un título. Tal como se ha mencionado previamente, el servidor de comercio electrónico 1400 se puede acoplar o bien a través de una red de área local privada o bien a través de una red de área global, tal como Internet con un servidor de procesamiento de crédito de terceros de un banco u otra institución financiera que puede realizar servicios tales como compensación de tarjetas de crédito, cargo en cuenta electrónica, etc. El módulo de extremo frontal 1404 y el módulo de extremo posterior 1410 del servidor 1400 se comunican a través de una serie de secuencias de comandos que están escritas de acuerdo con la norma de Interfaz de Pasarela Común (CGI, *Common Gateway Interface*). Será obvio a los razonablemente expertos en la materia que se pueden utilizar otras aplicaciones de servidor de comercio electrónico disponibles en el mercado con el sistema de SCDP de la presente invención además de las que se mencionan en el presente documento.

La base de datos 204, que está asociada con el servidor 202, puede comprender una base de datos serie convencional y se usa para almacenar información de crédito y de facturación necesaria para proseguir las transacciones.

El módulo de extremo frontal 1408 del servidor 1400 comprende adicionalmente el código o los objetos necesarios para generar cadenas de inicio tal como se explica con mayor detalle con referencia a la figura 9. Una vez que se han generado, las cadenas de inicio se reenvían al servidor CAS para la firma digital de las mismas.

En la realización ilustrativa, el sistema de comercio electrónico comprende un servidor y el escaparate electrónico, que funcionan conjuntamente para posibilitar que el usuario navegue a través de un catálogo y acepte y valide la información de compra. El sistema de comercio electrónico usa una arquitectura basada en web abierta para la interconexión con componentes externos. Los módulos de soporte lógico del sistema de SCDP de la presente invención se comunican con el soporte lógico de comercio electrónico mediante la publicación de URL al extremo frontal de servidor de web del soporte lógico de comercio electrónico. En respuesta a la publicación, Transact realiza una llamada a un programa de CGI con argumentos específicos que están codificados en el URL. La evaluación del URL por medio de la llamada de CGI da lugar a que el soporte lógico Transact cambie el estado de la base de datos. La totalidad de una secuencia de transacción se completa simplemente mediante la evaluación de un conjunto de URL. El sistema de comercio electrónico capturará y mantendrá datos de cliente, tales como cuentas de usuario o información de tarjetas de crédito.

Suponiendo que el sistema de comercio electrónico es un sistema completo que proporciona la capacidad de habilitar para comercio un escaparate electrónico y llevar a cabo transacciones de tarjetas de crédito a través de la web, la interacción entre el CAS y el sistema de comercio electrónico tiene lugar principalmente en tres lugares diferentes. Cuando el usuario ha comprado un título, al usuario se le presenta una página, a la que se hace referencia como “Recibo Digital”, en la que aparece un enlace que se denomina el URL de Cumplimiento. El URL de Cumplimiento es en realidad un programa de CGI cuyo fin es la obtención de una cadena de inicio a partir del CAS. Tal como se describe con mayor detalle en el presente documento, una cadena de inicio es una recopilación de la totalidad de la información que se necesita para que el CAS reconozca posteriormente el derecho del usuario al soporte lógico y, entonces, resuelva una transacción con el sistema de comercio electrónico. Esta información se devuelve en una forma que solo puede reconocer el CAS, de tal modo que el CAS puede validar posteriormente sus propias cadenas de inicio. La devolución de la cadena de inicio al navegador de cliente desencadena el navegador para activar el iniciador dentro del cliente de SCDP y pasa al iniciador la cadena de inicio. Posteriormente, el iniciador puede proporcionar una cadena de inicio al CAS y solicitar un activador. El CAS verifica la cadena de inicio y pide al servidor de comercio electrónico que valide que esta compra, si se resolviera, tendría éxito. No obstante, en la práctica el CAS no ha resuelto aún la transacción. Llegados a este punto, el CAS devuelve un activador al iniciador y se puede comenzar a ejecutar el título. El activador inicial se crea con un corta tiempo medio de vida, por ejemplo, por último, cuando el activador inicial está a punto de expirar, el VxD de cliente de SCDP notifica el iniciador y solicita que el CAS renueve el activador. Con la primera renovación del activador, la CASLIB32 proporciona de nuevo la cadena de inicio y, esta vez, el CAS resolverá la transacción con el servidor de comercio electrónico. El retardo de la resolución de la transacción permite que el sistema de SCDP garantice con seguridad que el título se ha ejecutado de forma apropiada en la máquina de cliente de SCDP antes de facturar su uso.

El sistema de SCDP soporta cinco modelos de compra diferentes. El primer modelo de compra, Título Los abonos ofrecen un acceso ilimitado a un título específico durante un periodo de tiempo especificado. Los abonos se pueden renovar. El segundo modelo de compra, Abonos a Paquete tales como un “Paquete de Juegos de Arcade”, ofrece un acceso ilimitado a un conjunto de múltiples títulos durante un tiempo limitado. El conjunto de títulos cubiertos por un abono a paquete podría cambiar con el tiempo. Por ejemplo, si el usuario compra un abono al “Paquete de Juegos Nuevos de Moda”, puede que los títulos disponibles en este paquete no sean los mismos que una semana o dos después de la compra de abono inicial. El tercer modelo de compra, Pago Por Uso, ofrece acceso una vez durante una cantidad ilimitada de tiempo. En el cuarto modelo de compra, Facturación Basada en Tiempo, se cobra más a un usuario por ejecutar el título durante más tiempo o este puede comprar un bloque fijo de tiempo. En el quinto modelo de compra, Facturación Mensual, el sistema de SCDP está integrado en un sistema de facturación de empresa de telecomunicaciones u Operación en Múltiples Servidores (MSO, *Multiple Server Operation*) por cable existente y añade cargos a la factura mensual del cliente. Se pueden añadir modelos de compra adicionales con cambios menores.

Escaparate electrónico virtual

El servidor de Escaparate Electrónico Virtual 215 y la base de datos 213 adjunta presentan un catálogo virtual a los clientes y clientes potenciales del sistema de SCDP 200. En la realización ilustrativa, el servidor 215 se puede implementar como un servidor web convencional, por ejemplo, una aplicación de servidor que se está ejecutando encima de un sistema operativo que, a su vez, se ejecuta encima de un soporte físico de servidor, similar a los que se describen con referencia al servidor de comercio electrónico 202. La aplicación de escaparate electrónico incluye una interfaz gráfica de usuario que presenta una serie de selecciones para que los clientes las examinen con un navegador de red convencional. Tales selecciones pueden incluir el nombre de un título particular, una breve descripción del título, opciones de compra o costes asociados, en el caso de un título multimedia, tal como una película o clip de audio, muestras breves del contenido del título, etc. Además, con cada selección de títulos está asociado un URN correspondiente. En ese sentido, el escaparate electrónico implementa el motor de consulta de base de datos apropiado para interactuar con la base de datos 213 en la que se puede almacenar el título, la descripción, los precios, la oferta digital y la información de URN para un gran número de posibles títulos dentro del sistema de SCDP 200. En respuesta a la selección de un título particular, la lógica de aplicación de escaparate

electrónico consulta la base de datos 213 en busca del URN correspondiente y reenvía la información apropiada al servidor de comercio electrónico 202 de una forma que se describe en el presente documento.

En la realización ilustrativa, el servidor de escaparate electrónico virtual 215 y la base de datos 213 están acoplados con el servidor de memoria caché 210 a través de una red de área local privada y segura 205, tal como se ha descrito en lo que antecede. Será obvio, no obstante, a los razonablemente expertos en la materia que el sistema de SCDP 200 se puede implementar con uno o más escaparates electrónicos virtuales que están acoplados con el servidor de memoria caché 210 y el servidor de comercio electrónico 202 a través de algo que no sea una red de área local, por ejemplo, una red de área global, tal como Internet de una forma comprendida de manera razonable por los expertos en la materia. En tales implementaciones, en las que el servidor de escaparate electrónico reside en una red pública, varios subconjuntos de información se pueden encontrar disponibles para su visualización por los clientes en perspectiva. Por ejemplo, los clientes que pagan una tasa de abono pueden tener acceso a un servidor de escaparate electrónico en la red privada que puede proporcionar más información y / o muestras de datos de título que el público general que accede a un servidor de escaparate electrónico que está ubicado en una red pública que puede proporcionar solo información mínima en relación con un título y sus opciones de compra asociadas.

Formato de BRIQ

La figura 12 ilustra de forma conceptual un diagrama de bloques de un BRIQ de acuerdo con la presente invención y sus componentes constituyentes. Tal como se ilustra, un BRIQ 1200 comprende un encabezado de BRIQ 1202, un bloque criptográfico 1204, un superbloque 1206 y uno o más títulos 1208A-1208N. El encabezado de BRIQ 1202 contiene una información que es usada por el módulo de iniciador dentro del cliente de SCDP, incluyendo una información tal como información de registro de sistema, resolución, título de aplicación, un URL, etc. El bloque criptográfico 1204 es usado por el VxD de ARFSD dentro del cliente de SCDP para determinar si el título está cifrado y, de ser así, la versión de clave criptográfica que se usa para tal cifrado. El superbloque 1206 puede incluir una información general acerca del BRIQ incluyendo el tamaño del BRIQ, la fecha de creación, la entrada en la que se puede hallar el directorio RAÍZ, etc. Cada uno de los títulos 1208A-N puede incluir un directorio y uno o más archivos que están asociados con un título particular. Tal como se explica en lo sucesivo en el presente documento, los BRIQ se almacenan en el servidor de RAFT, al que accede de forma remota un cliente de SDCP usando el protocolo de RAFT, y se presentan al sistema operativo del anfitrión como un sistema de archivos local.

De acuerdo con la presente invención, uno o más títulos se procesan y se empaquetan en forma de BRIQ, tal como se describe con referencia a la figura 12. El proceso mediante el cual se da formato de BRIQ a un título es tal como sigue. En primer lugar, una herramienta de utilidad, tal como la utilidad de visualización en el sistema operativo Windows se usa para extraer información de registro a partir de un título. Tales entradas de registro pueden comprender un conjunto mínimo de información tal como los nombres de archivo, los nombres de directorio y citas de configuraciones necesarios para ejecutar un título particular. Las entradas de registro extraídas se colocan en un archivo. A continuación, el archivo que contiene las entradas de registro se proporciona a un programa de creación. El programa de creación, en la realización ilustrativa, comprende código capaz de tomar los datos que comprenden el título y el archivo de entradas de registro y de cifrar tal información de acuerdo con cualquier número de algoritmos de cifrado disponibles en la actualidad. Los archivos cifrados resultantes se pueden almacenar en una jerarquía de directorios convencional, tal como se ilustra por medio de los directorios 1208A-N de la figura 12. A continuación, el directorio raíz del sistema de archivos y cualquier metainformación adicional incluyendo el tamaño del sistema de archivos, etc., se almacenan en el superbloque 206 del BRIQ 1200, tal como se ilustra en la figura 12. A continuación, una información acerca de la clave de descifrado, necesaria para descifrar la información cifrada dentro del BRIQ, se almacena en el bloque criptográfico 1204. La información dentro del bloque criptográfico puede comprender unos datos que identifican la versión de clave y una descripción del tipo de cifrado que se usa. La información en el bloque criptográfico 1204 puede estar parcialmente cifrada. Información tal como el URL de BRIQ, y requisitos de sistema se colocan en el encabezado de BRIQ 1202 junto con los nombres de los archivos y títulos ejecutables, y una correspondencia de la unidad de red y etiquetas adicionales. La información que está contenida dentro del encabezado de BRIQ 1202 no está cifrada, tal como se ilustra en la figura 12.

Activador

El activador tiene un formato tal como se ilustra en la figura 13. En concreto, un activador 1300 comprende un testigo 1302, unos datos de autorización 1304, una clave criptográfica 1306 y, opcionalmente, uno o más códigos de bytes 1308-1312. En la realización ilustrativa, el testigo 1302 se puede implementar un testigo de RAFT similar 800, tal como se ha descrito en lo que antecede con referencia a la figura 8 en el presente documento. Los datos de autorización 1304 comprenden los datos “de conexión persistente” útiles por el cliente de SCDP cuando se solicita un nuevo activador a partir del servidor CAS. Tales datos de autorización se pueden implementar con un código o cadena numérica simple o, como alternativa, pueden tener una implementación más sofisticada, tal como un valor de troceo de datos que estaban previamente asociados con el cliente. La clave 306 comprende datos criptográficos útiles en el descifrado de los datos que están contenidos dentro del BRIQ antes de la ejecución. Los datos criptográficos que comprenden la clave 306 pueden comprender una cadena de bits que es extraída por el intérprete de códigos de bytes 308 y se suministra al VxD de RAFT para facilitar el descifrado de datos de BRIQ.

En una realización simple, el activador 1300 comprende solo el testigo 1302, los datos de autorización 1304 y la clave 1306. En una realización más sofisticada, también se incluyen uno o más códigos de bytes 1308-1312 como parte del activador. En la realización ilustrativa, los códigos de bytes son, en esencia, instrucciones ejecutables en una máquina o bien física o bien virtual, tal como se implementa dentro del intérprete de códigos de bytes 308. En la realización ilustrativa, el intérprete de códigos de bytes 308 comprende una máquina virtual capaz de ejecutar tales códigos de bytes tal como se suministran al mismo a partir del activador. El tipo y la naturaleza de los posibles códigos de bytes 1 - N que se pueden usar con el activador 1300 se describen en lo sucesivo en el presente documento. El intérprete de códigos de bytes 308 se describe con referencia a la figura 3C.

Ofuscación de código

La esencia de un programa se puede descomponer en flujo y primitivas. Por lo general, el flujo incluye la construcción de abstracciones de más alto nivel a partir de primitivas. La optimización comporta combinar primitivas redundantes, reordenando el flujo de tal manera que se puedan combinar y eliminar estructuras similares, y reconocer patrones y sustituir estos con otros patrones más eficientes. La optimización preserva el comportamiento de un programa con respecto a la especificación original. Las operaciones de ofuscación pueden producir más de un resultado correcto. En lugar de realizar la selección de forma aleatoria, puede ser más eficiente en conjunto producir la totalidad o un cierto subconjunto de variantes correctas en paralelo, a un cierto coste para la producción individual.

De forma genérica, la optimización comporta la búsqueda de formas de tomar una solución para un problema y modificar la misma para producir una solución mejor. En concreto durante la compilación, esta implica tomar una expresión correcta y producida de forma simple de un fragmento de código de alto nivel y convertir la misma en un código más eficiente al tiempo que se preserva su corrección. La pesimización también preserva esta corrección, pero sacrifica eficiencia por dificultad de descompilación en forma de ofuscación.

Mediante la división de las fases de extremo frontal y de ensamblador se permite la inserción de pesimizadores a diferentes niveles y permite lenguajes de alto nivel alternativos posteriores (tal como Lisp / Scheme) que prevén más flexibilidad en la pesimización. Se puede usar un número de técnica de pesimizador, incluyendo 1) Pesimizador de Mirilla de nivel Ensamblador que toma flujos de códigos de bytes y realiza reordenación local y ofuscación; 2) Pesimizador de Lenguaje intermedio que expone la capa de traducción entre el lenguaje de alto nivel y el ensamblador con el fin de proporcionar una interfaz más natural para determinadas pesimizaciones estructurales; y 3) Pesimizador Manual de Alto nivel que, en lugar de realizar en la práctica unas operaciones genéricas sobre código de lenguaje de alto nivel, permite que el codificador especifique múltiples formas de expresar una función dada y, entonces, hacer que el compilador produzca directamente un formulario con la expansión combinatoria de alternativas ya iniciadas.

En teoría, siempre es posible que alguien ejecute en una única etapa el activador y supervise los cambios que realiza este, y entienda por lo tanto cómo descodificar el BRIQ, o incluso más simplemente, que pare una vez que el BRIQ se ha descodificado y vuelque el texto no cifrado fuera de la memoria. Mediante el uso de diferentes secuencias de códigos de bytes, escritas de una forma “ofuscada” difícil de interpretar, y al evitar la reutilización de unas idénticas, por ejemplo claves maestras, la presente invención utiliza unas construcciones que hacen difícil el trabajo del descompilador humano, e imposible el análisis automático. El código de bytes hace que el trabajo de un desencriptador no autorizado en una única descarga consuma mucho tiempo de una manera arbitraria, y que no sea aplicable a ninguna otra descarga. Las técnicas de ofuscación de muestras útiles con los activadores pueden incluir

- Seleccionar de entre un gran grupo de algoritmos para cada operación de tal modo que incluso una segunda solicitud del mismo objeto obtiene un código significativamente diferente;
- Aplicar operaciones de preservación de comportamiento directamente al código de bytes, usando por ejemplo técnicas de optimización de compilador.
- Hacer que el cliente de SCDP soporte múltiples conjuntos de códigos de bytes, o que codifique de forma criptográfica la propia lista de códigos de bytes.
- Un código de bytes automodificable.
- Flujos de códigos de bytes de “trampilla”, por ejemplo que generan una secuencia de códigos de bytes, y una función de puesta en correspondencia que escoge un subconjunto y pone en correspondencia el objeto de código de bytes con un algoritmo útil. Puede que sea necesario definir restricciones y, entonces, buscar secuencias útiles en un espacio.
- Códigos de bytes de “código no alcanzado”, posiblemente relacionados por patrón con códigos existentes como una distracción.
- “Abstenerse de” determinados códigos de bytes, por ejemplo, el código tiene diferentes significados en ejecuciones posteriores (las herramientas de alto nivel pueden simplemente intercalar algoritmos de trabajo para producir los mismos). Las extensiones incluyen abstenerse de toda instrucción que haga referencia a una ubicación o registro particular.
- Operadores “unarios” para su uso en implementaciones criptográficas.
- Optimizar la puesta en correspondencia de códigos de bytes basándose en parámetros del código, por ejemplo, la frecuencia de uso, factores no relacionados, etc.
- Cuando se implementa cifrado directamente en un código de bytes, distribuir secuencias de código o clave / programas parciales para generar claves en lugar de claves de formato “convencional”.

- Hacer que el código de bytes descargue un código de bytes adicional a través de devoluciones de llamada posteriores, o hacer que el servidor envíe cambios de códigos de bytes de forma asíncrona.
- Usar datos de entorno existentes como fuentes de códigos de bytes, datos, material de generación de claves o entropía débil, tal como el propio BRIQ, u otros archivos binarios en el entorno, o incluso el código de bytes descargado.

Lo ideal es que las ofuscaciones se puedan producir en un marco de trabajo que proporcione información acerca de cómo se pueden combinar entre sí las mismas, y cómo se pueden realizar operaciones sobre estas.

Técnicas

Otra forma de dar a los Activadores fuerza adicional es hacerlos incompletos, de tal modo que es necesario que estos establezcan un contacto adicional con el CAS para continuar funcionando. Una “Técnica” es un fragmento de código que se ejecuta en el CAS y está personalizado para soportar tales solicitudes. A pesar de que se podrían usar múltiples técnicas, una técnica singular puede atender a una clase de activadores. Una implementación de Técnica simple se puede codificar de forma rígida en el CAS o, como alternativa, implementarse con objetos compartidos o un código de bytes cargado de forma dinámica. El activador para el protocolo de Técnica puede ser una capa encima de una RPC existente para el transporte procedente del cliente de SCDP, eliminando la necesidad para la técnica de tener mensajes previamente definidos. El código de bytes de activador y el código de bytes de Técnica se pueden tratar como lenguajes diferenciados. En su lugar, el código de Técnica puede simplemente tener unos códigos de bytes singulares para rutinas criptográficas enteras.

Con el fin de implementar un código de bytes ofuscado en los activadores, se utilizan los siguientes componentes: 1) un intérprete de códigos de bytes; 2) un ensamblador de códigos de bytes; 3) unas rutinas de códigos de bytes criptográficas; 4) una interfaz con el VxD de ARFS para llamar al activador en puntos útiles; 5) un protocolo tal como se describe en el presente documento que posibilita que el activador se comunique con la implementación de la técnica en el CAS; 6) unas funciones de construcción de Activador de CAS (la fábrica de activadores 710);

El lector apreciará que el sistema de la presente invención que se describe en el presente documento facilita la distribución a petición de contenido seguro a través de redes de banda ancha así como intranets privadas.

La invención que se ha descrito en lo que antecede se puede implementar o bien completamente en soporte lógico, o bien completamente en soporte físico, o bien en una combinación de soporte físico y de soporte lógico, incluyendo código de programa que se almacena en formato de soporte lógico inalterable para soportar un soporte físico dedicado. Una implementación en soporte lógico de la realización o realizaciones que se han descrito en lo que antecede puede comprender una serie de instrucciones informáticas o bien fijas en un medio tangible, tal como un medio legible por ordenador, por ejemplo el disquete 142, el CD ROM 147, la ROM 115 o el disco fijo 152 de la figura 1, o bien transmisibles a un sistema informático en una onda portadora, por medio de un módem u otro dispositivo de interfaz, tal como el adaptador de comunicaciones 190 que está conectado con la red 195 a través de un medio 191. El medio 191 o bien puede ser un medio tangible, incluyendo pero sin limitarse a líneas de comunicaciones ópticas o analógicas, o bien se puede implementar con técnicas inalámbricas, incluyendo pero sin limitarse a microondas, infrarrojos u otras técnicas de transmisión. La serie de instrucciones informáticas, ya esté contenida en un medio tangible o en una onda portadora, materializa la totalidad o parte de la funcionalidad que se ha descrito en lo que antecede en el presente documento con respecto a la invención. Los expertos en la materia apreciarán que tales instrucciones informáticas se pueden escribir en un número de lenguajes de programación para su uso con muchas arquitecturas informáticas o sistemas operativos y pueden existir en un formato ejecutable por máquina. Además, tales instrucciones se pueden almacenar usando cualquier tecnología de memoria, presente o futura, incluyendo pero sin limitarse a, dispositivos de memoria de semiconductores, magnéticos, ópticos o de otros tipos, o transmitirse usando cualquier tecnología de comunicaciones, presente o futura, incluyendo pero sin limitarse a tecnologías de transmisión ópticas, de infrarrojos, de microondas, o de otros tipos. Se contempla que un producto de programa informático de este tipo se pueda distribuir como un medio extraíble con documentación impresa o electrónica adjunta, por ejemplo, soporte lógico retractilado, precargado con un sistema informático, por ejemplo, en una ROM de sistema o un disco fijo, o distribuido a partir de un servidor o panel de anuncios electrónicos a través de una red, por ejemplo, Internet o World Wide Web.

A pesar de que se han divulgado varias realizaciones a modo de ejemplo de la invención, será evidente a los expertos en la materia que se pueden realizar varios cambios y modificaciones que lograrán algunas de las ventajas de la invención sin apartarse del ámbito de la invención. Será obvio a los razonablemente expertos en la materia que otros componentes que realicen las mismas funciones se pueden sustituir de forma conveniente. Además, los procedimientos de la invención de pueden lograr o bien completamente en implementaciones de soporte lógico, usando las instrucciones de procesador apropiadas, o bien en implementaciones híbridas que utilizan una combinación de lógica de soporte físico y lógica de soporte lógico para lograr los mismos resultados.

REIVINDICACIONES

1. Un procedimiento de ejecución de una aplicación en un sistema informático local sin que la aplicación esté instalada en el sistema informático local, comprendiendo el procedimiento:

montar (600) un sistema de archivos de red remota de un sistema informático remoto en el sistema informático local en el que el sistema informático local tiene acceso al sistema de archivos de red remota montado y en el que la aplicación se almacena en el sistema de archivos de red remota;
almacenar (602) entradas de registro asociadas con la aplicación en un área de memoria local del sistema informático local;
ejecutar (604) la aplicación almacenada en el sistema de archivos de red remota montado bajo el control de un sistema operativo en el sistema informático local;
interceptar (606) llamadas de acceso a registro procedentes del sistema operativo que ejecuta la aplicación; y
redirigir (608) llamadas de acceso a registro interceptadas a las entradas de registro almacenadas.

2. El procedimiento de la reivindicación 1, que comprende adicionalmente desmontar (612) el sistema de archivos de red remota montado tras la finalización (610) de la ejecución de la aplicación.

3. El procedimiento de la reivindicación 1, que comprende adicionalmente eliminar las entradas de registro almacenadas del área de memoria local del sistema informático local tras la finalización de la ejecución de la aplicación.

4. El procedimiento de la reivindicación 1, que comprende adicionalmente:

recibir información de sistema de archivos asociada con la aplicación;
procesar la información de sistema de archivos recibida para identificar un atributo de escritura a través representativo de un archivo de datos para almacenar información después de la ejecución de la aplicación, en el que el archivo de datos se almacena en un dispositivo local asociado con el sistema informático local.

5. El procedimiento de la reivindicación 4, en el que el procesamiento comprende adicionalmente generar un sistema de archivos en el sistema informático local basándose en el atributo de escritura identificado.

6. El procedimiento de la reivindicación 1, en el que un controlador de dispositivo intercepta las llamadas de acceso a registro.

7. Un sistema informático de ejecución de una aplicación en el sistema informático sin instalar la aplicación en el sistema informático, comprendiendo el ordenador un procesador, una memoria y un sistema operativo, comprendiendo el sistema informático:

una lógica de programa configurada para montar (600) un sistema de archivos de red remota de un sistema informático remoto en el sistema informático, en el que el sistema informático tiene acceso al sistema de archivos de red remota montado y en el que la aplicación se almacena en el sistema de archivos de red remota;
una lógica de programa configurada para almacenar (602) una pluralidad de entradas de registro en relación con la aplicación en una unidad física local del sistema informático;
una lógica de programa configurada para ejecutar (604) la aplicación almacenada en el sistema de archivos de red remota montado bajo el control del sistema operativo del sistema informático; y
una lógica de programa para interceptar (606) llamadas de acceso a registro procedentes del sistema operativo; y
una lógica de programa para redirigir (608) llamadas de acceso a registro interceptadas a la pluralidad almacenada de entradas de registro.

8. El sistema informático de la reivindicación 7, que comprende adicionalmente una lógica de programa configurada para desmontar (612) el sistema de archivos de red remota montado tras la finalización (610) de la ejecución de la aplicación.

9. El sistema informático de la reivindicación 7, que comprende adicionalmente una lógica de programa configurada para eliminar la pluralidad almacenada de entradas de registro de la unidad física local del sistema informático.

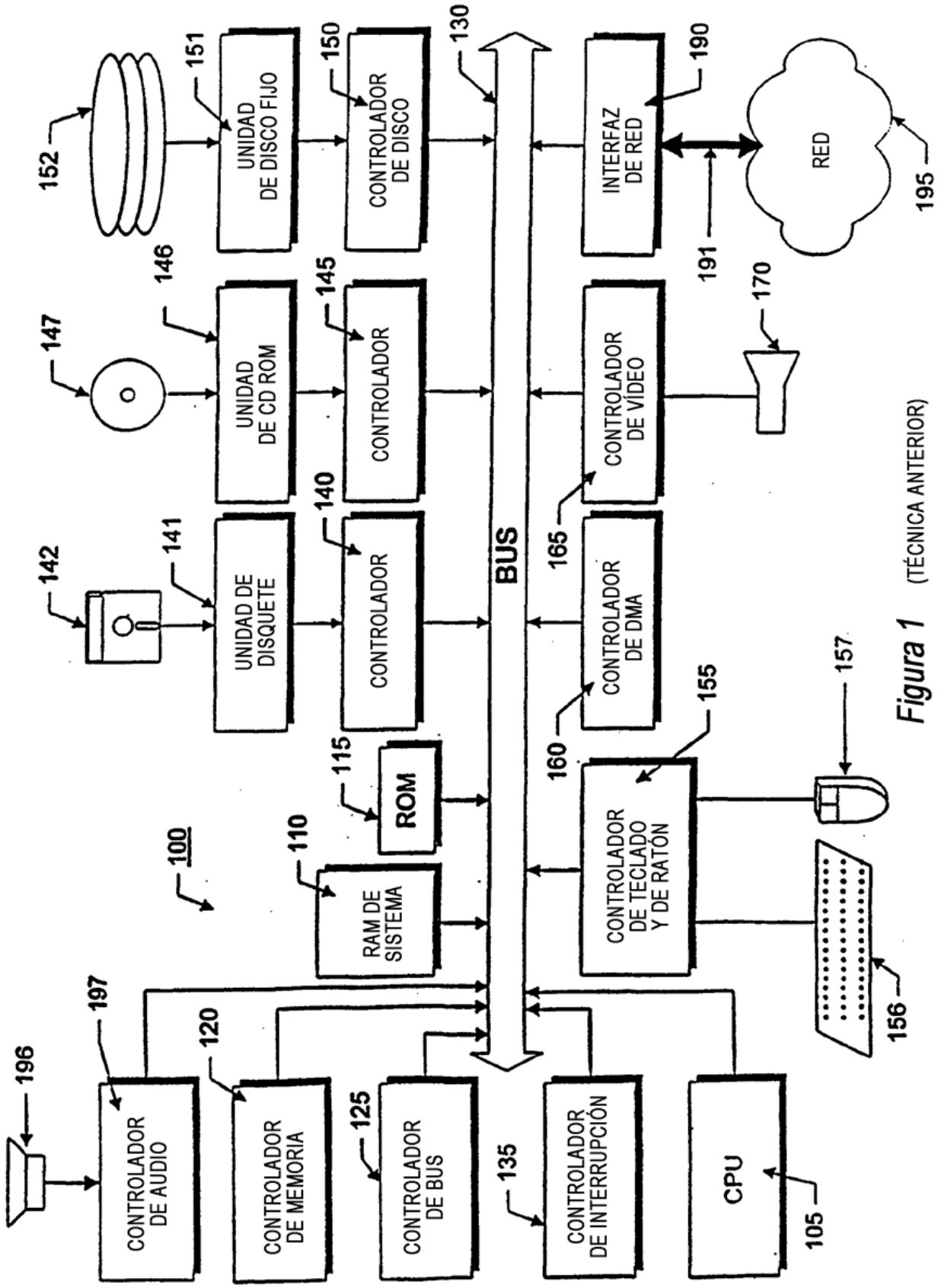

Figura 1 (TÉCNICA ANTERIOR)

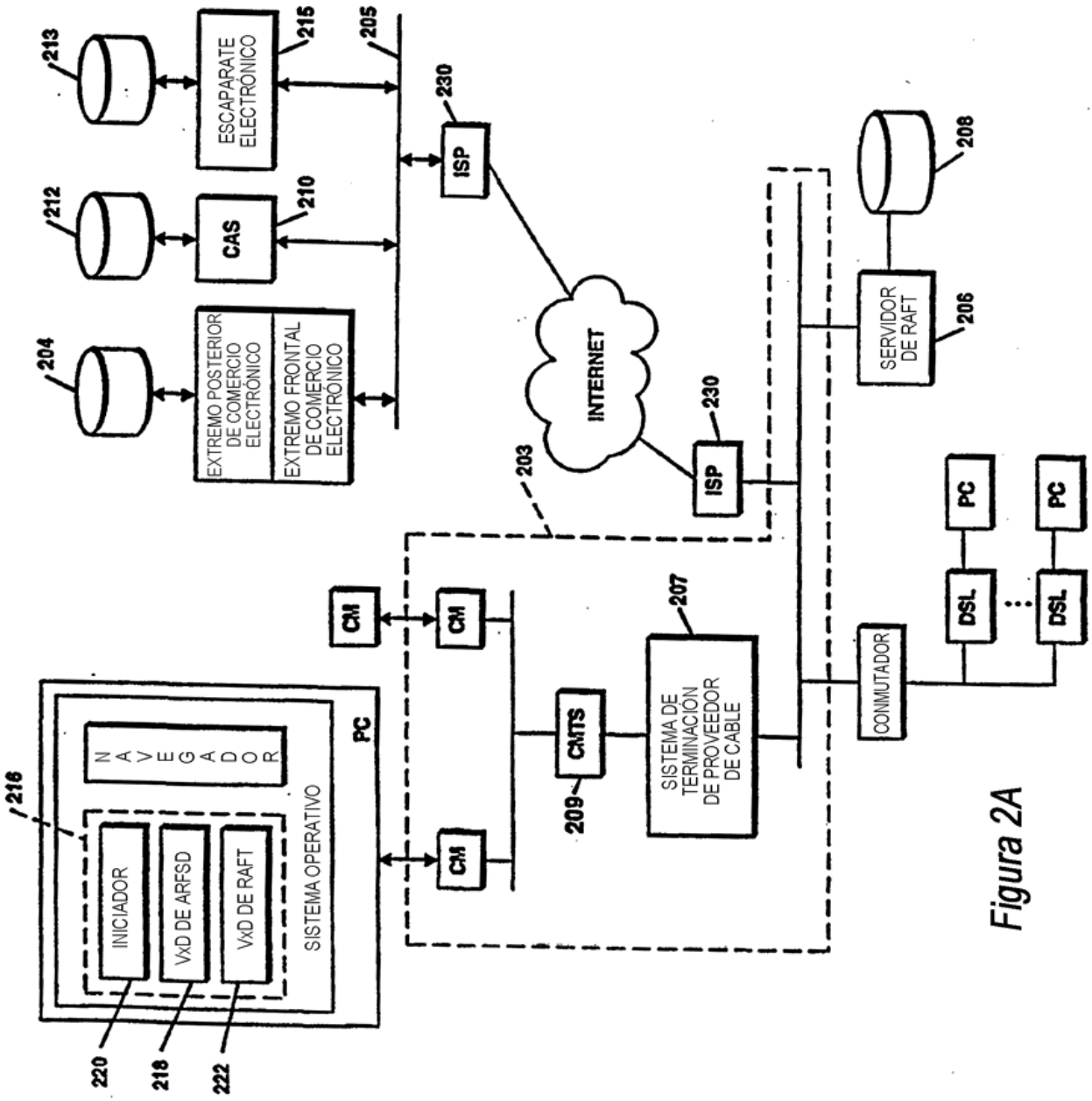
213
215
205
ESCAPARATE ELECTRÓNICO
230
ISP
212
CAS
210
208
204
EXTREMO POSTERIOR DE COMERCIO ELECTRÓNICO
EXTREMO FRONTAL DE COMERCIO ELECTRÓNICO
INTERNET
SERVIDOR DE RAFT
206
203
ISP
230
PC
PC
CM
CM
207
DSL
DSL
CMTS
209
SISTEMA DE TERMINACIÓN DE PROVEEDOR DE CABLE
CONMUTADOR
216
NAVEGADOR
INICIADOR
VxD DE ARFSD
VxD DE RAFT
SISTEMA OPERATIVO
PC
CM
220
218
222

Figura 2A

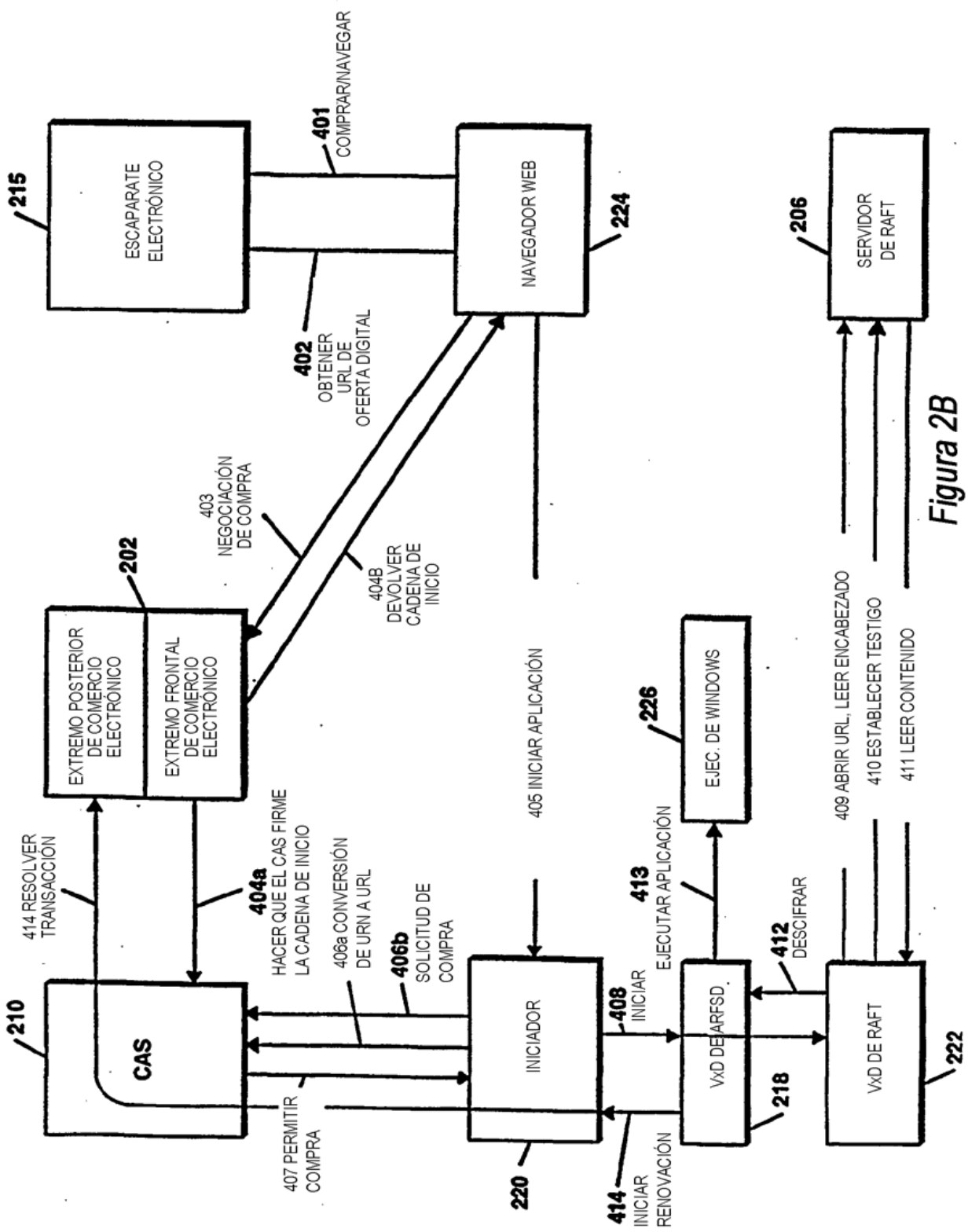
215
ESCAPARATE ELECTRÓNICO
401 COMPRAR/NAVEGAR
NAVEGADOR WEB
224
206
SERVIDOR DE RAFT
402 OBTENER URL DE OFERTA DIGITAL
403 NEGOCIACIÓN DE COMPRA
404B DEVOLVER CADENA DE INICIO
202
EXTREMO POSTERIOR DE COMERCIO ELECTRÓNICO
EXTREMO FRONTAL DE COMERCIO ELECTRÓNICO
405 INICIAR APLICACIÓN
226
EJEC. DE WINDOWS
409 ABRIR URL, LEER ENCABEZADO
410 ESTABLECER TESTIGO
411 LEER CONTENIDO
414 RESOLVER TRANSACCIÓN
404a HACER QUE EL CAS FIRME LA CADENA DE INICIO
406a CONVERSIÓN DE URN A URL
406b SOLICITUD DE COMPRA
413 EJECUTAR APLICACIÓN
412 DESCIFRAR
210
CAS
INICIADOR
408 INICIAR
VxD DE ARFSD
VxD DE RAFT
218
222
220
407 PERMITIR COMPRA
414 INICIAR RENOVACIÓN

Figura 2B

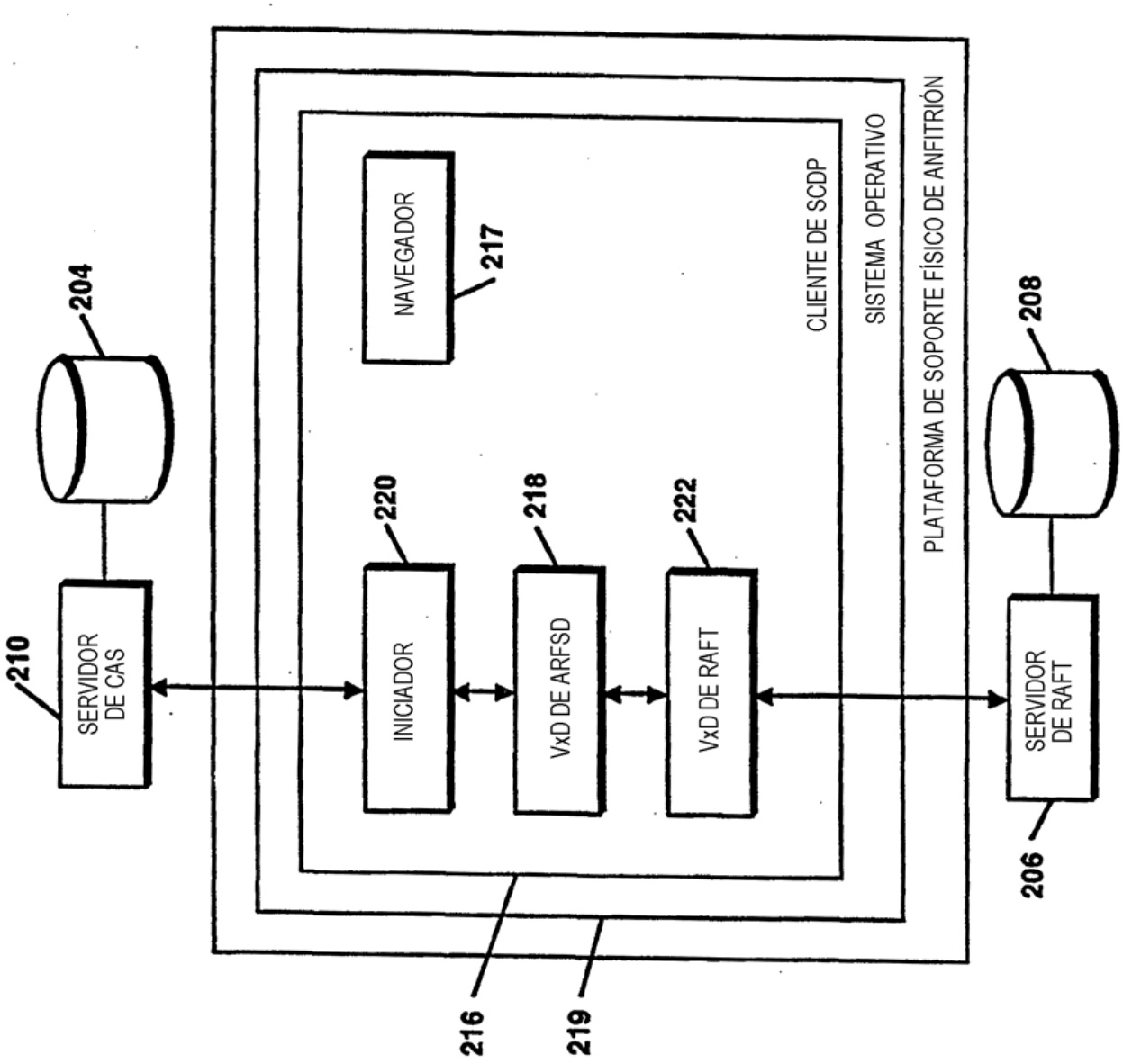
PLATAFORMA DE SOPORTE FÍSICO DE ANFITRIÓN
SISTEMA OPERATIVO
CLIENTE DE SCDP
NAVEGADOR
217
INICIADOR
220
VxD DE ARFSD
218
VxD DE RAFT
222
SERVIDOR DE CAS
210
SERVIDOR DE RAFT
206
204
208
216
219

Figura 3A

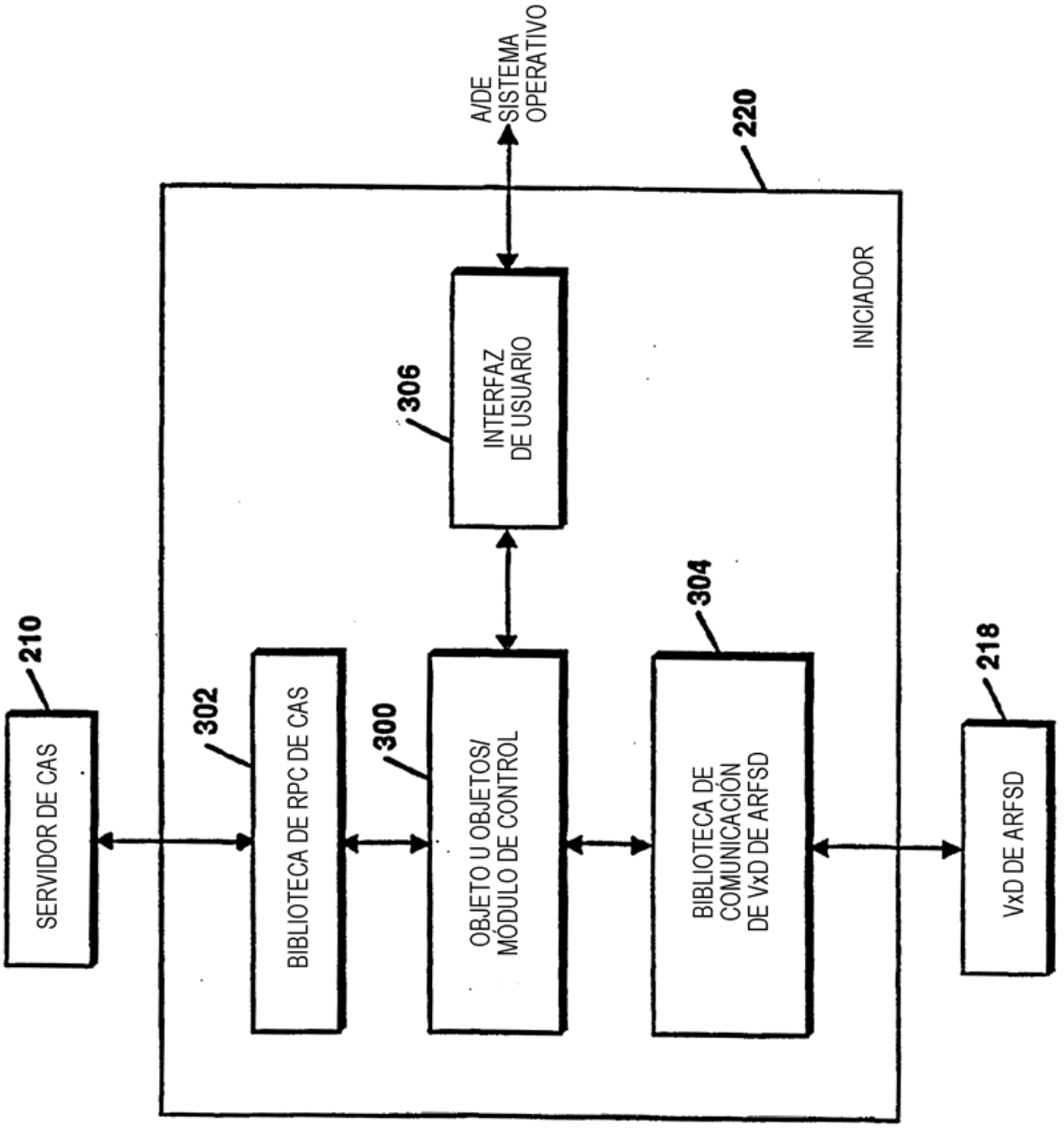
A/DE SISTEMA OPERATIVO
220
INICIADOR
306
INTERFAZ DE USUARIO
210
SERVIDOR DE CAS
302
BIBLIOTECA DE RPC DE CAS
300
OBJETO U OBJETOS/ MÓDULO DE CONTROL
304
BIBLIOTECA DE COMUNICACIÓN DE VxD DE ARFSD
218
VxD DE ARFSD

Figura 3B

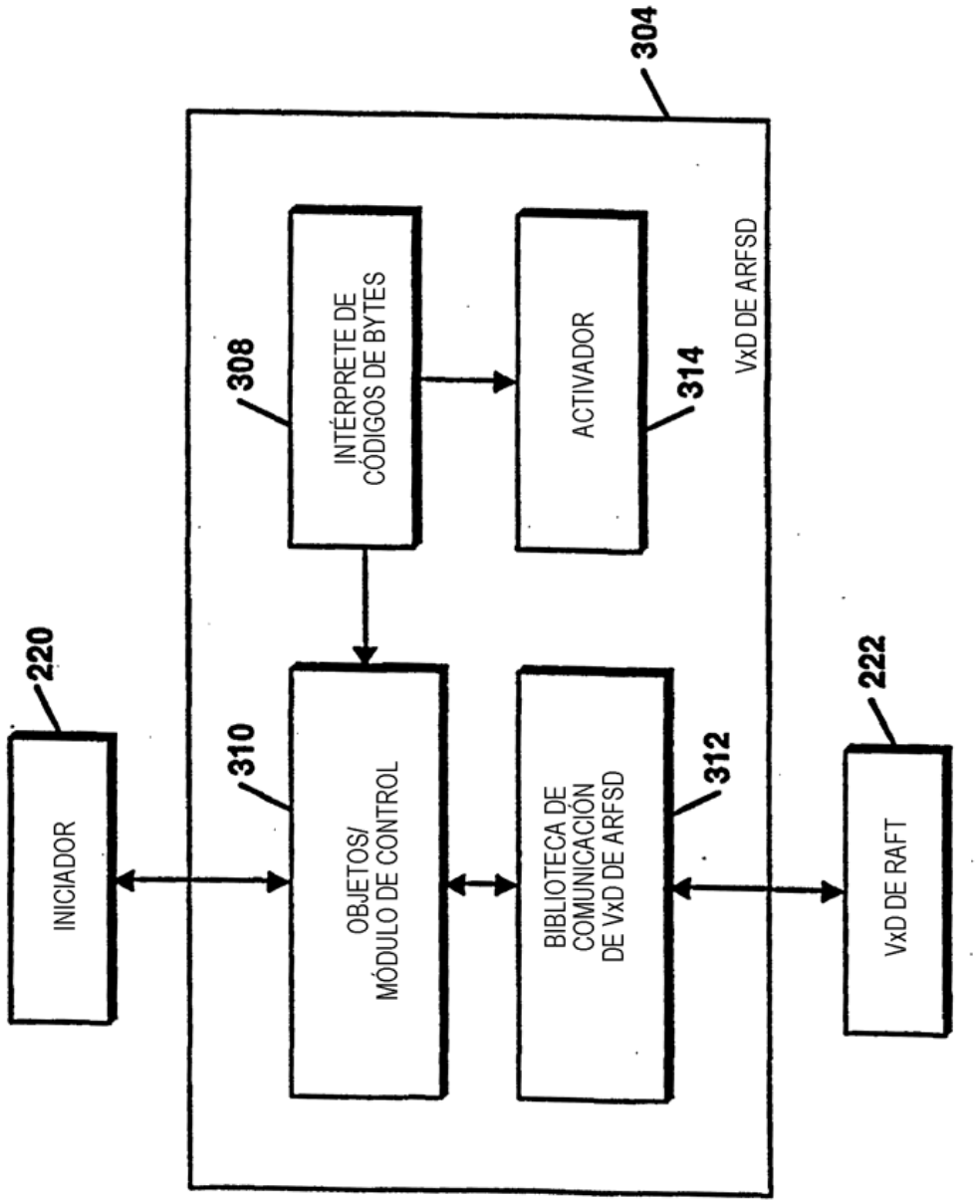
304
VxD DE ARFSD
308
INTÉRPRETE DE CÓDIGOS DE BYTES
ACTIVADOR
314
220
INICIADOR
310
OBJETOS/ MÓDULO DE CONTROL
BIBLIOTECA DE COMUNICACIÓN DE VxD DE ARFSD
312
222
VxD DE RAFT

Figura 3C

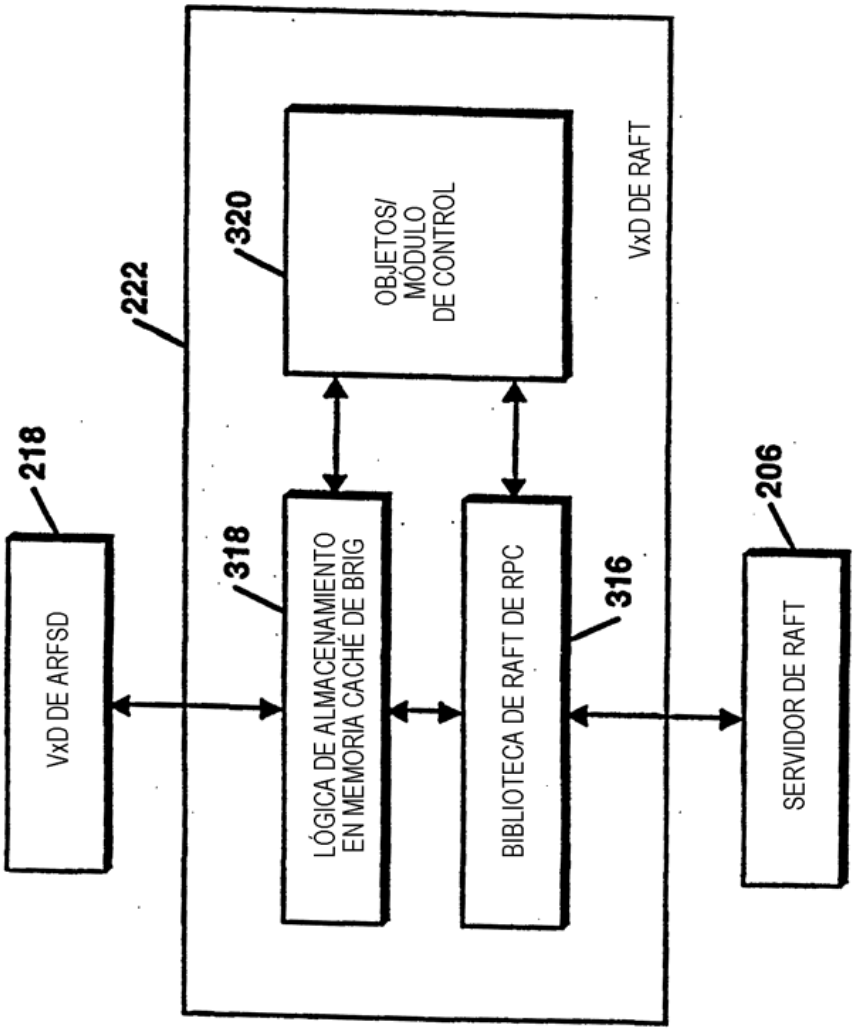
218
VxD DE ARFSD
222
318
LÓGICA DE ALMACENAMIENTO EN MEMORIA CACHÉ DE BRIG
320
OBJETOS/ MÓDULO DE CONTROL
VxD DE RAFT
316
BIBLIOTECA DE RAFT DE RPC
206
SERVIDOR DE RAFT

Figura 3D

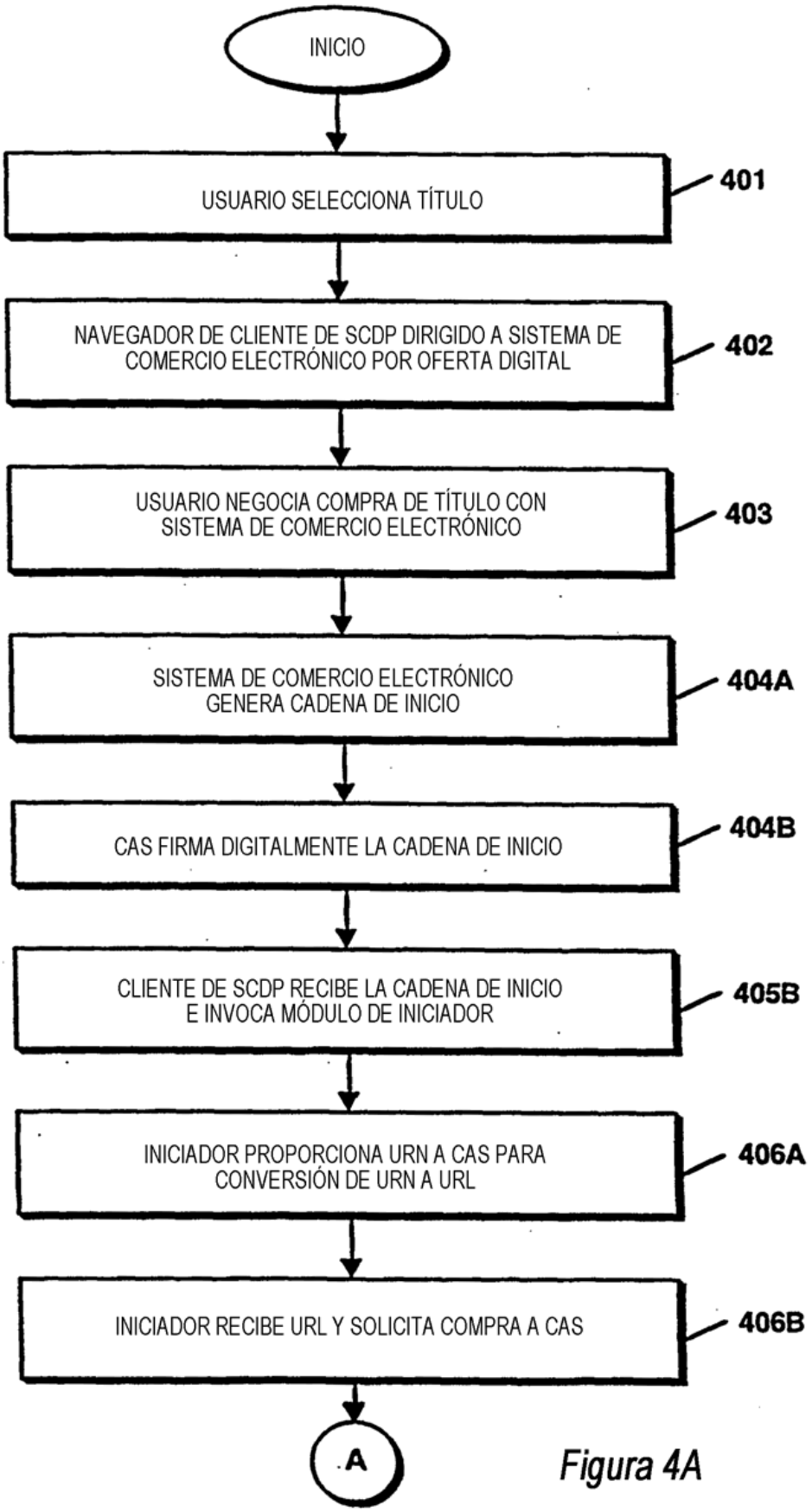
INICIO
USUARIO SELECCIONA TÍTULO
401
NAVEGADOR DE CLIENTE DE SCDP DIRIGIDO A SISTEMA DE COMERCIO ELECTRÓNICO POR OFERTA DIGITAL
402
USUARIO NEGOCIA COMPRA DE TÍTULO CON SISTEMA DE COMERCIO ELECTRÓNICO
403
SISTEMA DE COMERCIO ELECTRÓNICO GENERA CADENA DE INICIO
404A
CAS FIRMA DIGITALMENTE LA CADENA DE INICIO
404B
CLIENTE DE SCDP RECIBE LA CADENA DE INICIO E INVOCA MÓDULO DE INICIADOR
405B
INICIADOR PROPORCIONA URN A CAS PARA CONVERSIÓN DE URN A URL
406A
INICIADOR RECIBE URL Y SOLICITA COMPRA A CAS
406B
A

Figura 4A

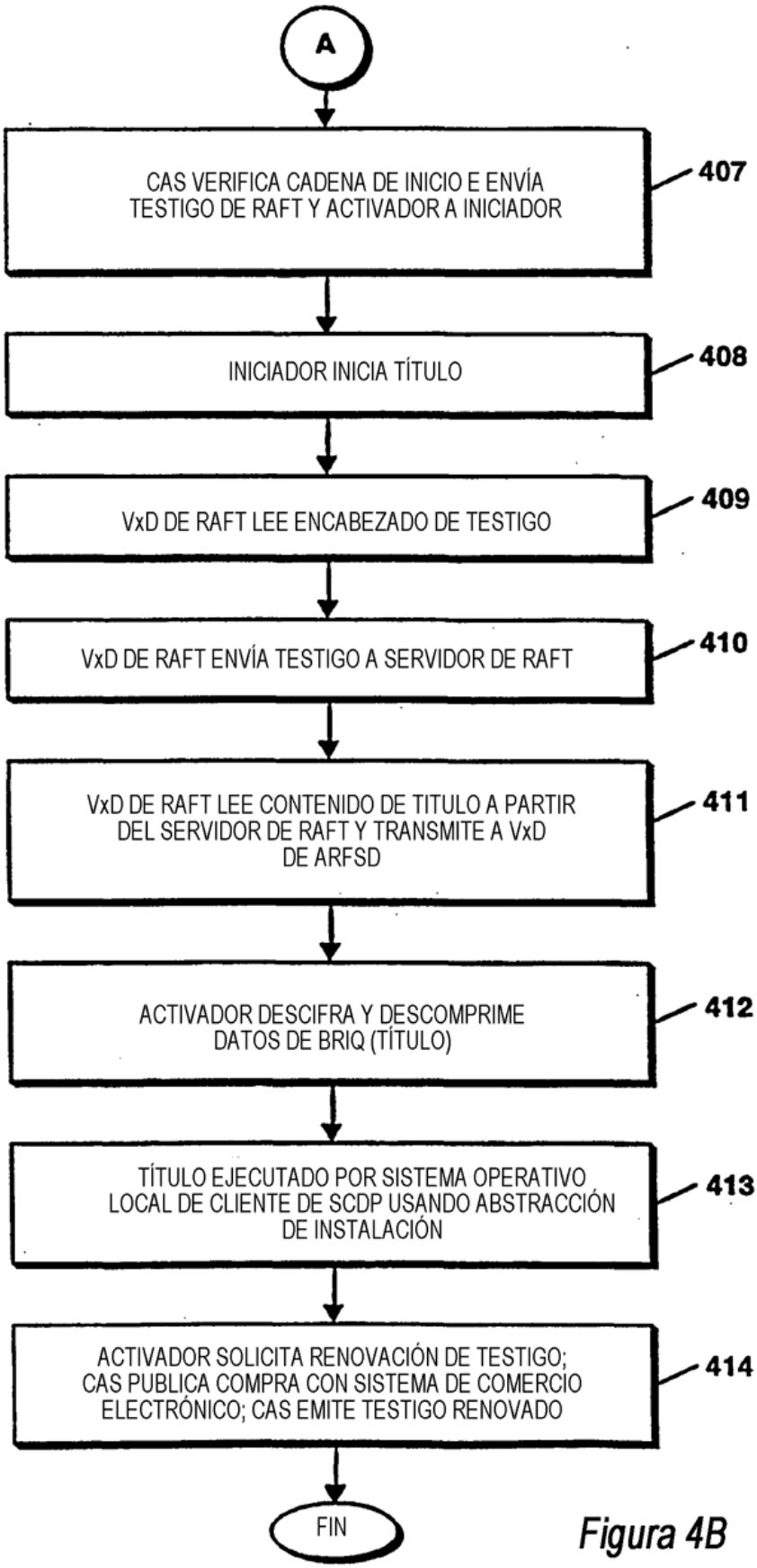
A
CAS VERIFICA CADENA DE INICIO E ENVÍA TESTIGO DE RAFT Y ACTIVADOR A INICIADOR
407
INICIADOR INICIA TÍTULO
408
VxD DE RAFT LEE ENCABEZADO DE TESTIGO
409
VxD DE RAFT ENVÍA TESTIGO A SERVIDOR DE RAFT
410
VxD DE RAFT LEE CONTENIDO DE TITULO A PARTIR DEL SERVIDOR DE RAFT Y TRANSMITE A VxD DE ARFSD
411
ACTIVADOR DESCIFRA Y DESCOMPRIME DATOS DE BRIQ (TÍTULO)
412
TÍTULO EJECUTADO POR SISTEMA OPERATIVO LOCAL DE CLIENTE DE SCDP USANDO ABSTRACCIÓN DE INSTALACIÓN
413
ACTIVADOR SOLICITA RENOVACIÓN DE TESTIGO; CAS PUBLICA COMPRA CON SISTEMA DE COMERCIO ELECTRÓNICO; CAS EMITE TESTIGO RENOVADO
414
FIN

Figura 4B

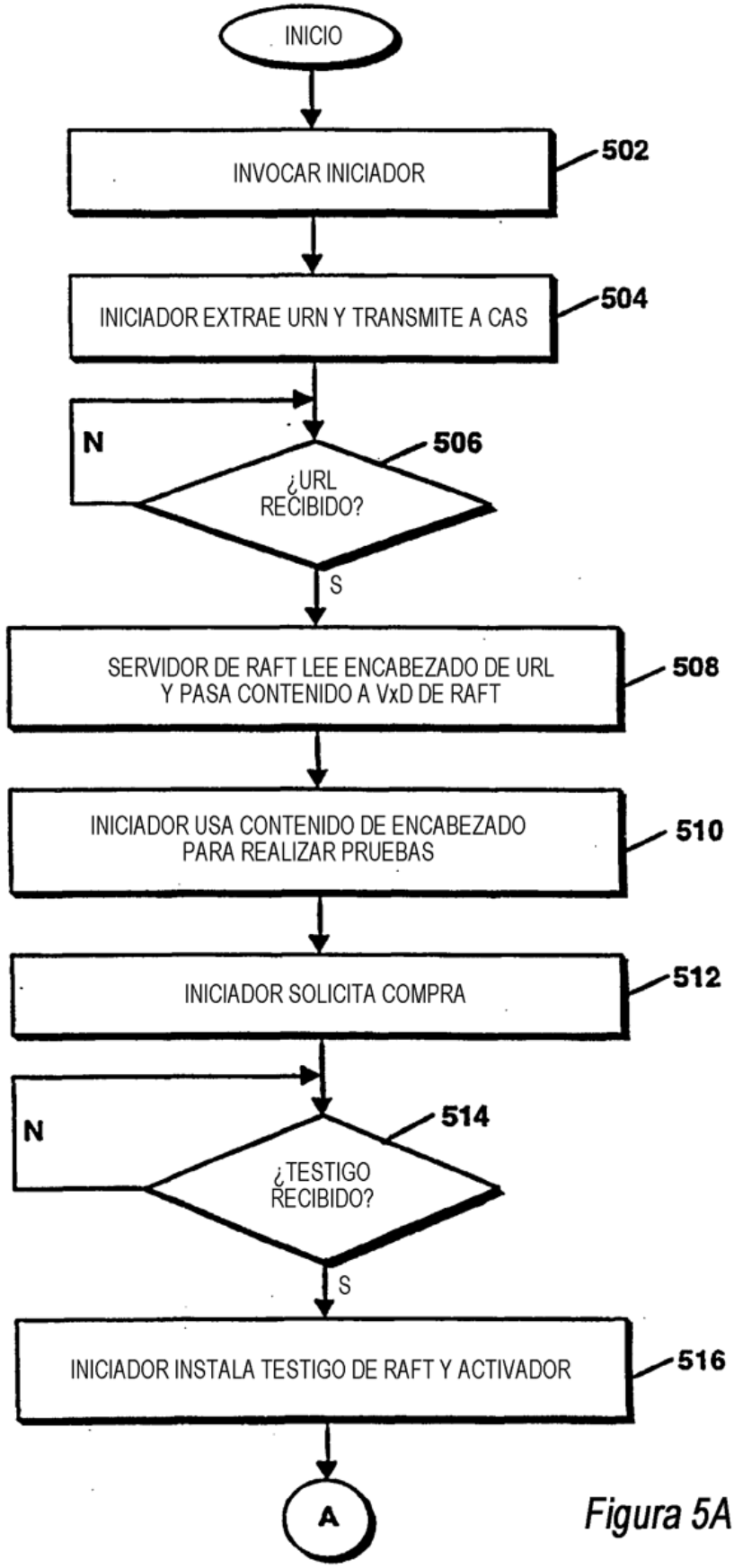
INICIO
INVOCAR INICIADOR
502
INICIADOR EXTRAE URN Y TRANSMITE A CAS
504
N
¿URL RECIBIDO?
506
S
SERVIDOR DE RAFT LEE ENCABEZADO DE URL Y PASA CONTENIDO A VxD DE RAFT
508
INICIADOR USA CONTENIDO DE ENCABEZADO PARA REALIZAR PRUEBAS
510
INICIADOR SOLICITA COMPRA
512
N
¿TESTIGO RECIBIDO?
514
S
INICIADOR INSTALA TESTIGO DE RAFT Y ACTIVADOR
516
A

Figura 5A

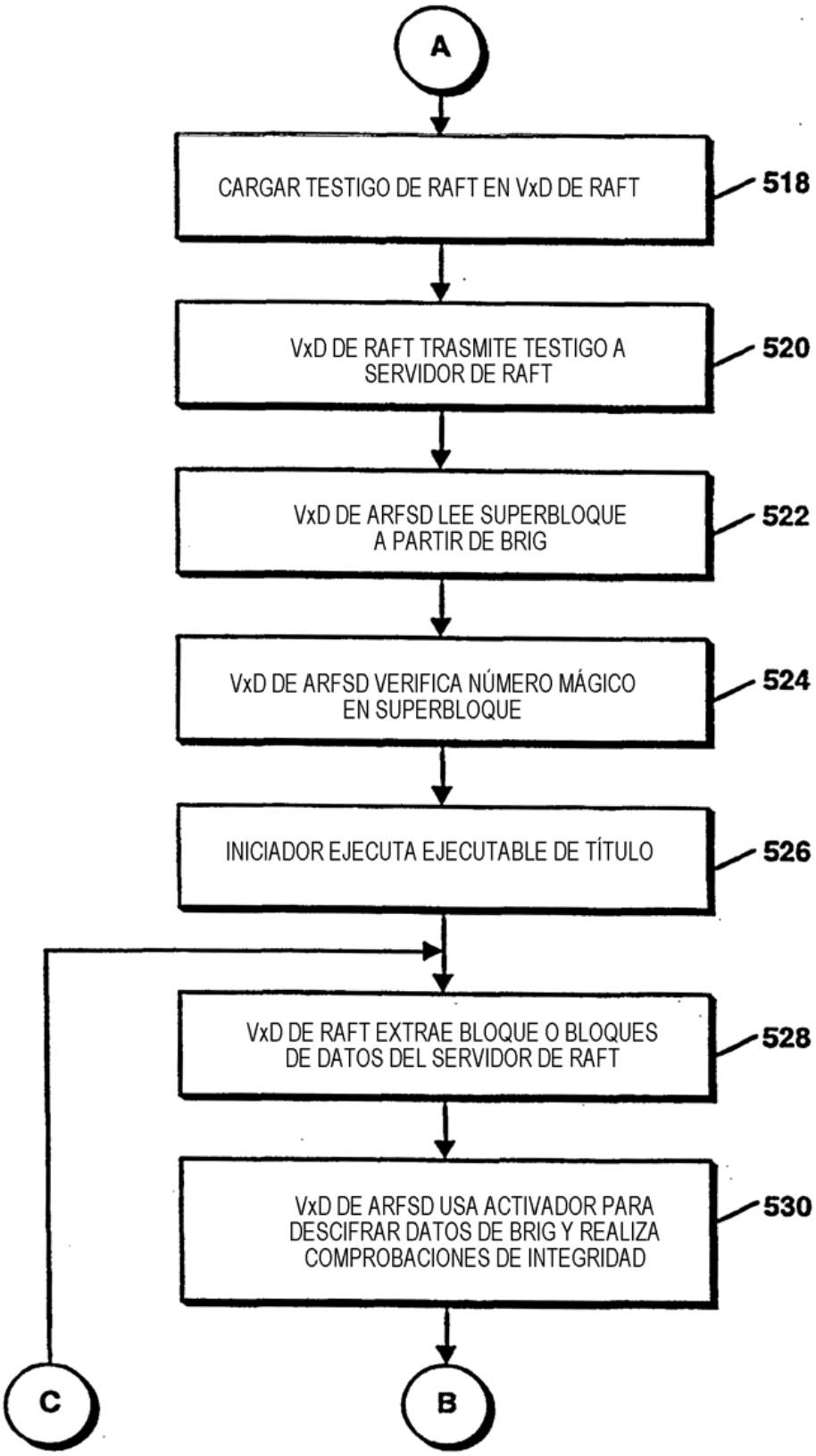
A
CARGAR TESTIGO DE RAFT EN VxD DE RAFT
518
VxD DE RAFT TRASMITE TESTIGO A SERVIDOR DE RAFT
520
VxD DE ARFSD LEE SUPERBLOQUE A PARTIR DE BRIG
522
VxD DE ARFSD VERIFICA NÚMERO MÁGICO EN SUPERBLOQUE
524
INICIADOR EJECUTA EJECUTABLE DE TÍTULO
526
VxD DE RAFT EXTRAE BLOQUE O BLOQUES DE DATOS DEL SERVIDOR DE RAFT
528
VxD DE ARFSD USA ACTIVADOR PARA DESCIFRAR DATOS DE BRIG Y REALIZA COMPROBACIONES DE INTEGRIDAD
530
C
B

Figura 5B

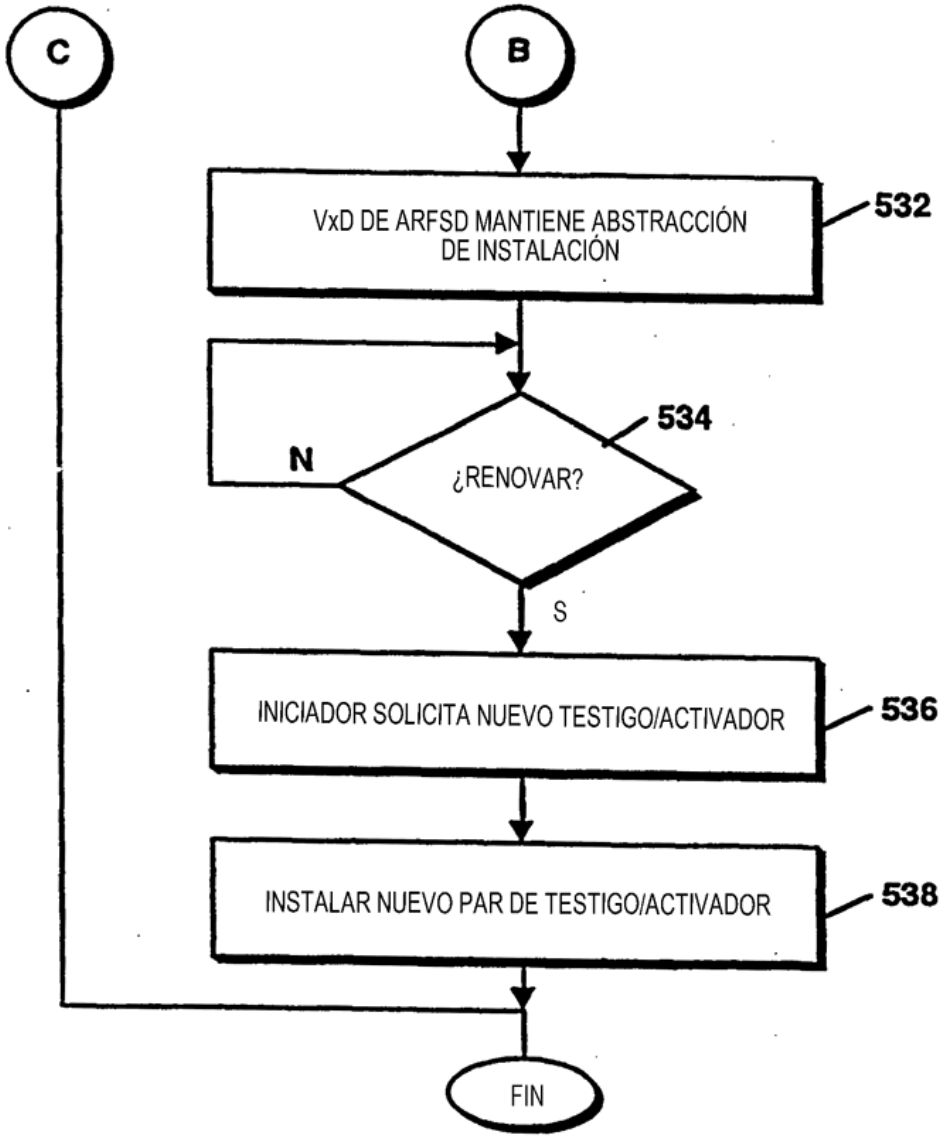
C
B
VxD DE ARFSD MANTIENE ABSTRACCIÓN DE INSTALACIÓN
532
534
N
¿RENOVAR?
S
INICIADOR SOLICITA NUEVO TESTIGO/ACTIVADOR
536
INSTALAR NUEVO PAR DE TESTIGO/ACTIVADOR
538
FIN

Figura 5C

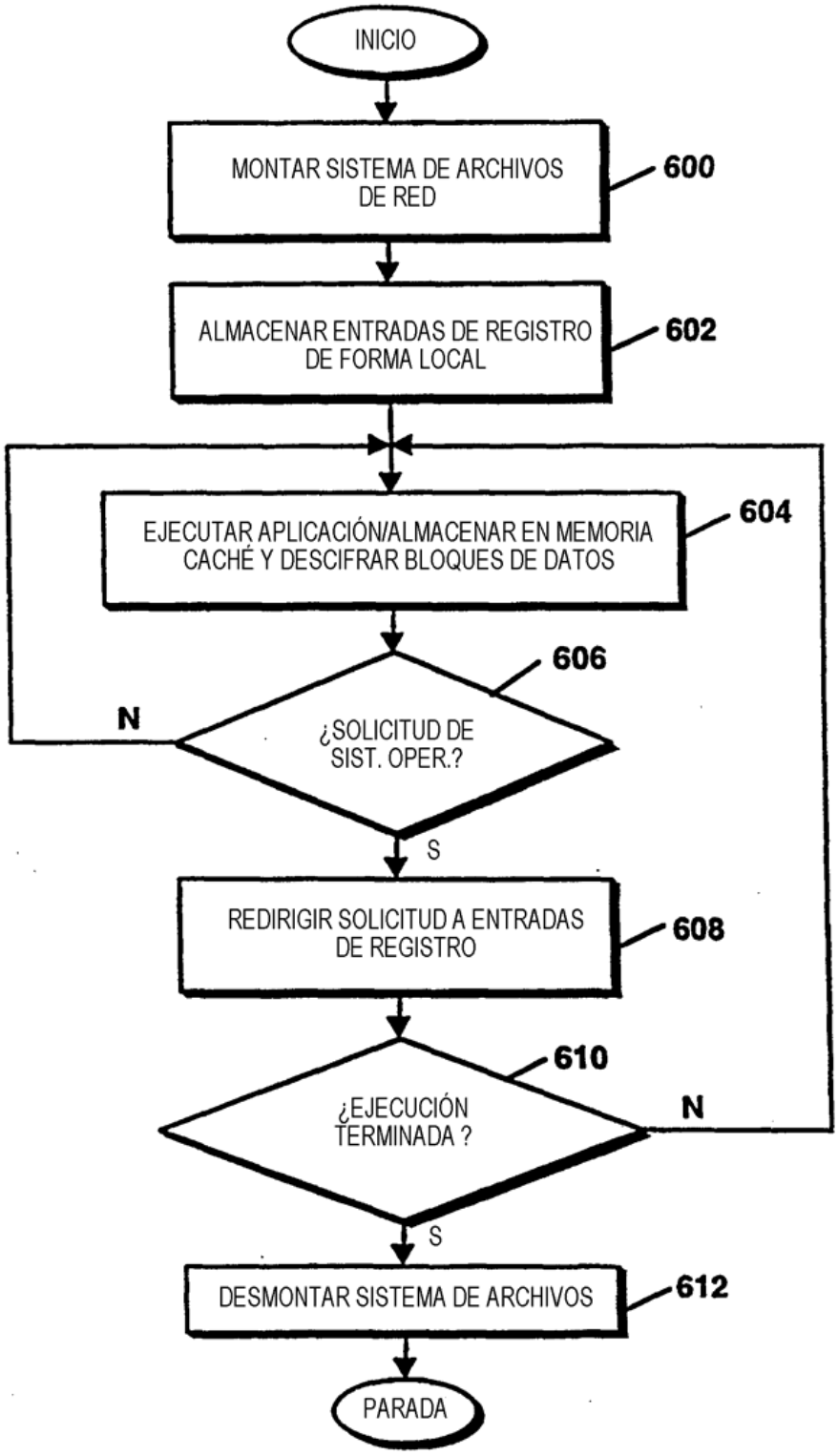
INICIO
MONTAR SISTEMA DE ARCHIVOS DE RED
600
ALMACENAR ENTRADAS DE REGISTRO DE FORMA LOCAL
602
EJECUTAR APLICACIÓN/ALMACENAR EN MEMORIA CACHÉ Y DESCIFRAR BLOQUES DE DATOS
604
606
¿SOLICITUD DE SIST. OPER.?
N
S
REDIRIGIR SOLICITUD A ENTRADAS DE REGISTRO
608
610
¿EJECUCIÓN TERMINADA ?
N
S
DESMONTAR SISTEMA DE ARCHIVOS
612
PARADA

Figura 6

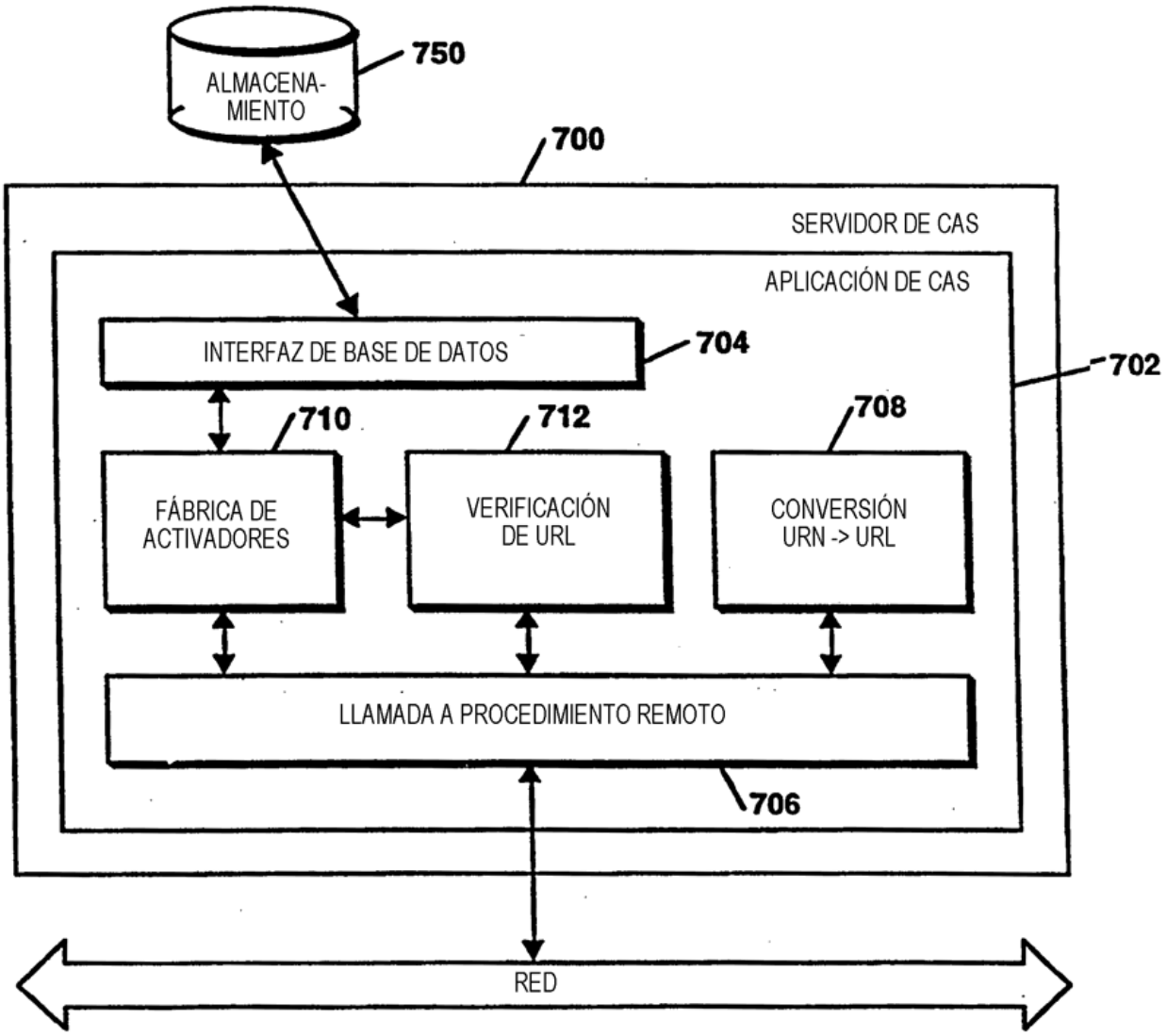
ALMACENA-
MIENTO
750
700
SERVIDOR DE CAS
APLICACIÓN DE CAS
INTERFAZ DE BASE DE DATOS
704
702
710
712
708
FÁBRICA DE
ACTIVADORES
VERIFICACIÓN
DE URL
CONVERSIÓN
URN -> URL
LLAMADA A PROCEDIMIENTO REMOTO
706
RED

Figura 7A

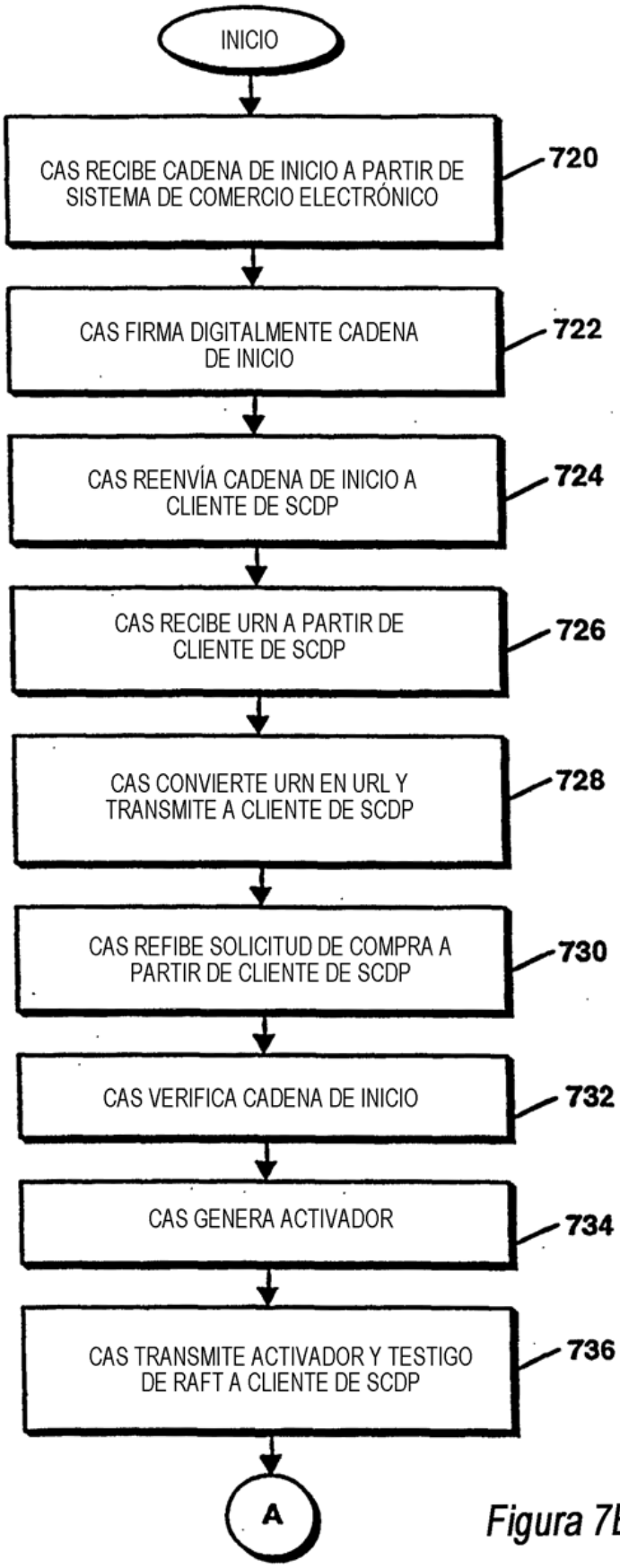
INICIO
CAS RECIBE CADENA DE INICIO A PARTIR DE SISTEMA DE COMERCIO ELECTRÓNICO
720
CAS FIRMA DIGITALMENTE CADENA DE INICIO
722
CAS REENVÍA CADENA DE INICIO A CLIENTE DE SCDP
724
CAS RECIBE URN A PARTIR DE CLIENTE DE SCDP
726
CAS CONVIERTE URN EN URL Y TRANSMITE A CLIENTE DE SCDP
728
CAS REFIBE SOLICITUD DE COMPRA A PARTIR DE CLIENTE DE SCDP
730
CAS VERIFICA CADENA DE INICIO
732
CAS GENERA ACTIVADOR
734
CAS TRANSMITE ACTIVADOR Y TESTIGO DE RAFT A CLIENTE DE SCDP
736
A

Figura 7B

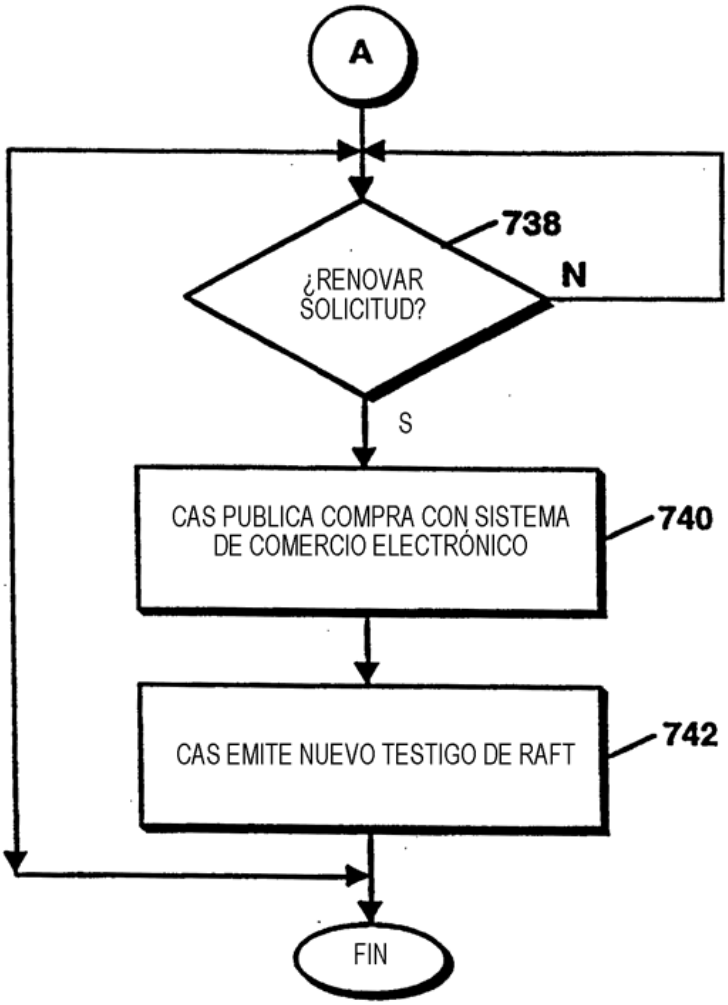

Figura 7B Cont

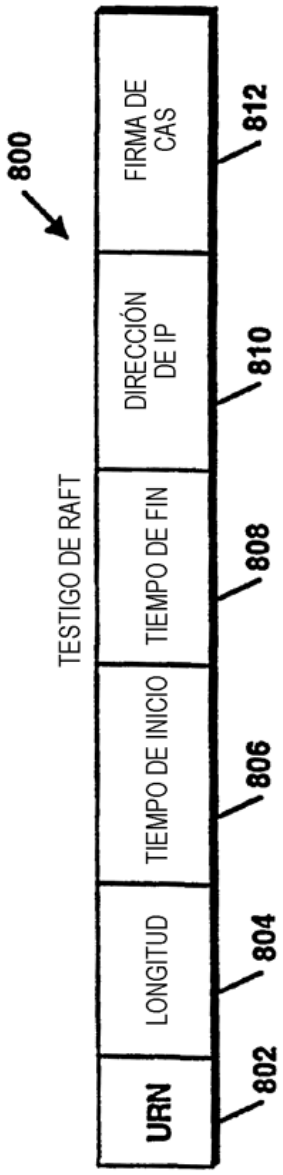

Figura 8

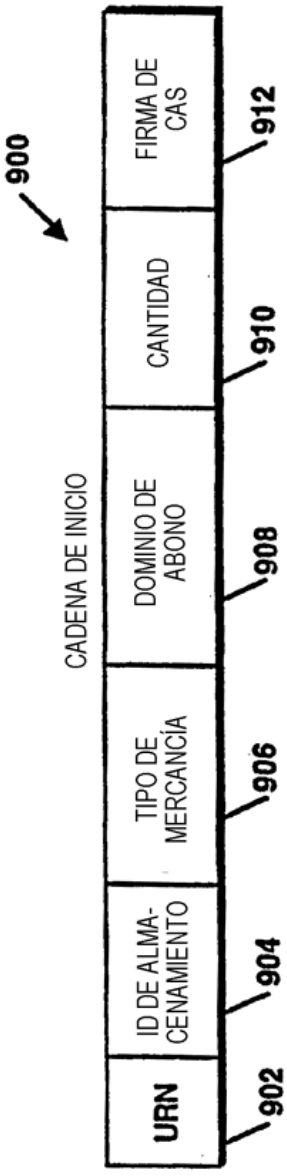

Figura 9

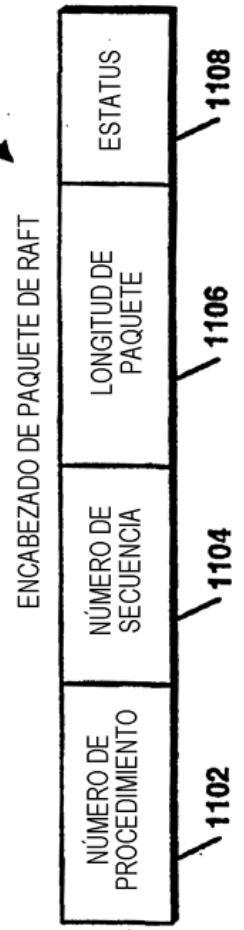

Figura 11

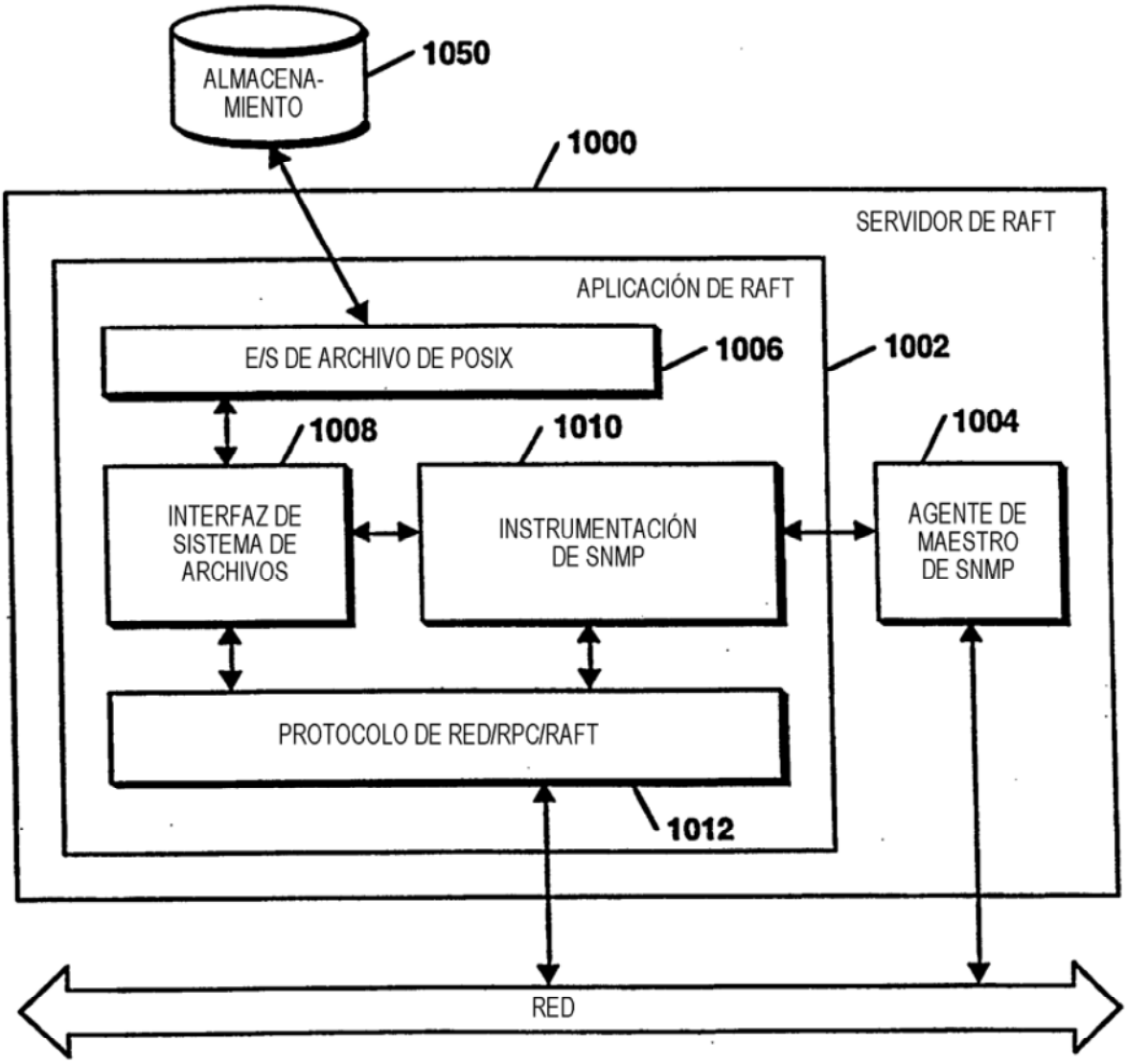
ALMACENA-
MIENTO
1050
1000
SERVIDOR DE RAFT
APLICACIÓN DE RAFT
E/S DE ARCHIVO DE POSIX
1006
1002
1008
1010
1004
INTERFAZ DE
SISTEMA DE
ARCHIVOS
INSTRUMENTACIÓN
DE SNMP
AGENTE DE
MAESTRO
DE SNMP
PROTOCOLO DE RED/RPC/RAFT
1012
RED

Figura 10

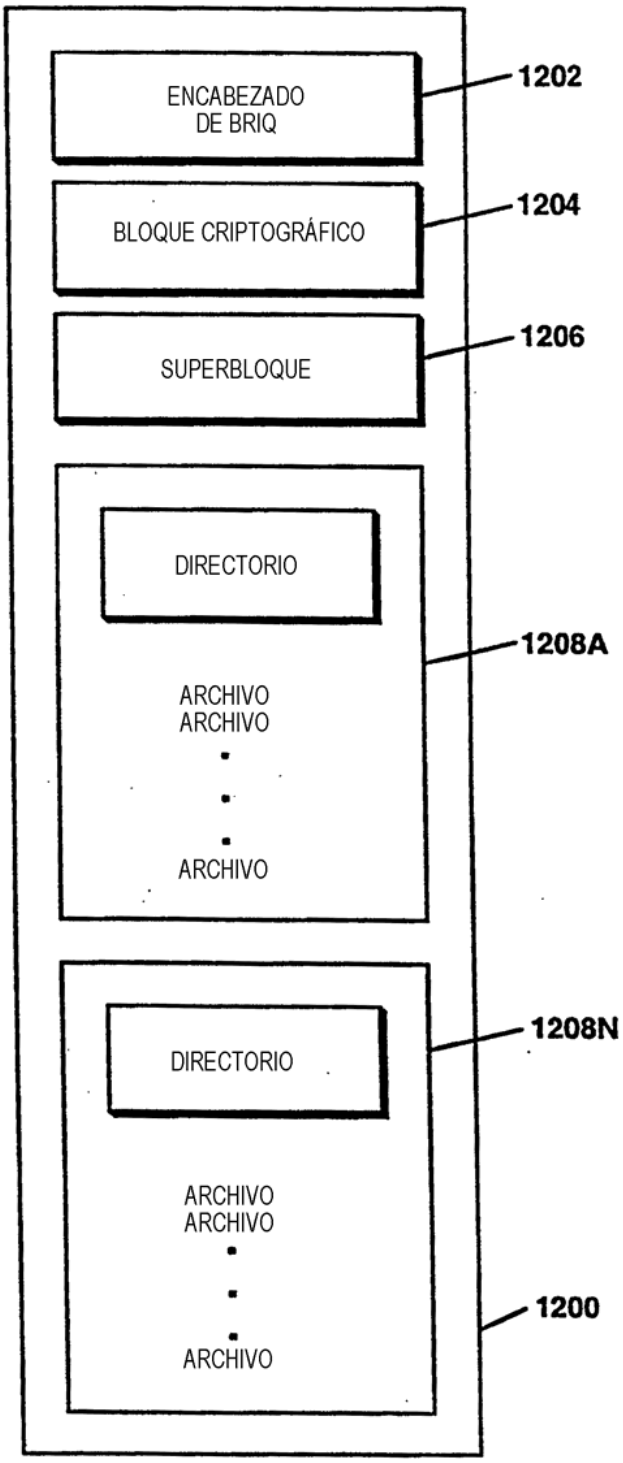
ENCABEZADO DE BRIQ
1202
BLOQUE CRIPTOGRÁFICO
1204
SUPERBLOQUE
1206
DIRECTORIO
ARCHIVO
ARCHIVO
ARCHIVO
1208A
DIRECTORIO
ARCHIVO
ARCHIVO
ARCHIVO
1208N
1200

Figura 12

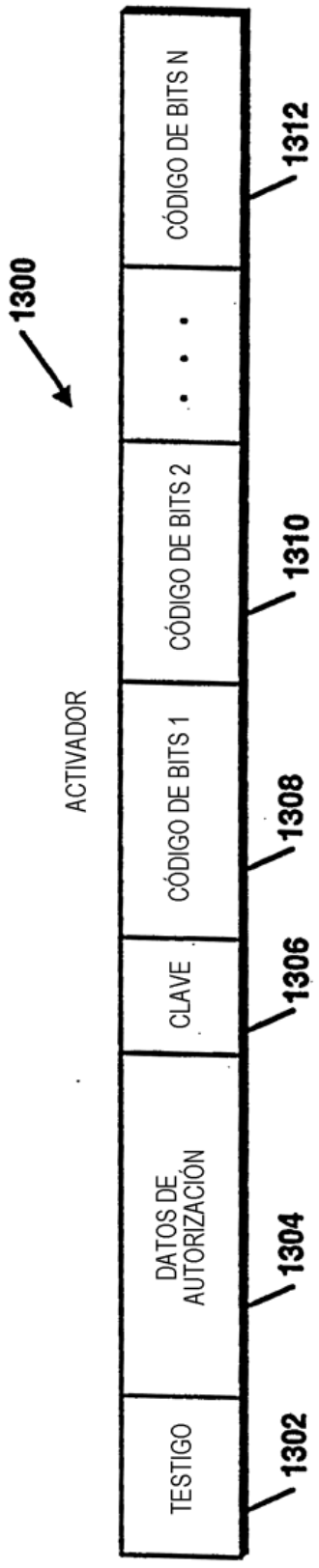
1300
ACTIVADOR
TESTIGO
1302
DATOS DE AUTORIZACIÓN
1304
CLAVE
1306
CÓDIGO DE BITS 1
1308
CÓDIGO DE BITS 2
1310
. . .
CÓDIGO DE BITS N
1312

Figura 13

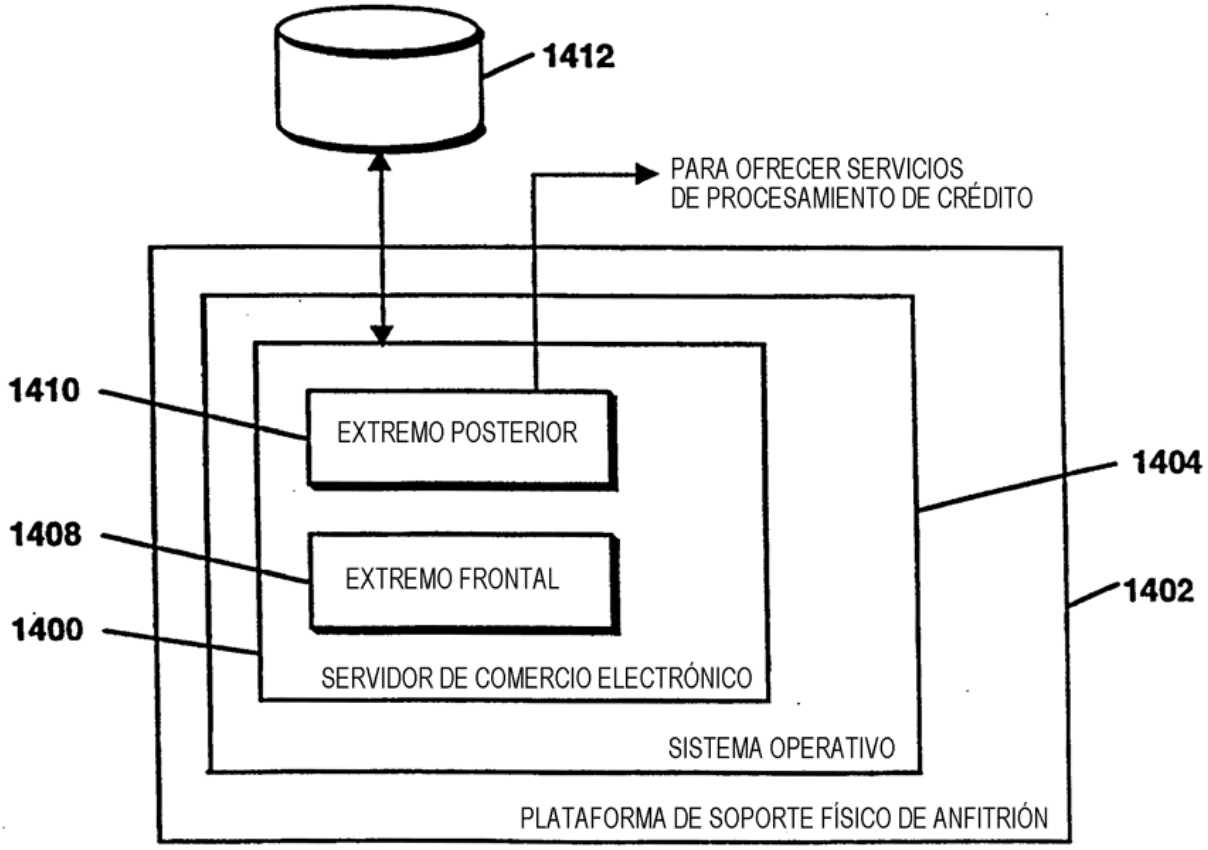
1412
PARA OFRECER SERVICIOS
DE PROCESAMIENTO DE CRÉDITO
1410
EXTREMO POSTERIOR
1404
1408
EXTREMO FRONTAL
1402
1400
SERVIDOR DE COMERCIO ELECTRÓNICO
SISTEMA OPERATIVO
PLATAFORMA DE SOPORTE FÍSICO DE ANFITRIÓN

Figura 14