(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 600 745**

(51) Int. CI.:

G06F 17/30 (2006.01)
G06F 7/00 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **09.03.2006 PCT/US2006/008416**

(87) Fecha y número de publicación internacional: **09.11.2006 WO06118662**

(96) Fecha de presentación y número de la solicitud europea: **09.03.2006 E 06737577 (4)**

(97) Fecha y número de publicación de la concesión europea: **03.08.2016 EP 1875389**

(54) Título: **Seguridad basada en la región**

(30) Prioridad:

04.05.2005 US 122299

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
10.02.2017

(73) Titular/es:

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

(72) Inventor/es:

LI, ZIQUAN y
DUTTA, TANMOY

(74) Agente/Representante:

CARPINTERO LÓPEZ, Mario

DESCRIPCIÓN

Seguridad basada en la región

Campo técnico

La presente invención se refiere en general a sistemas informáticos y, más particularmente, se refiere a sistemas y procedimientos que proporcionan seguridad a un subconjunto de objetos, en función de un descriptor de región para el subconjunto con el fin de mitigar los requisitos de propagación y de almacenamiento de datos de las jerarquías de herencia de objetos clásicos.

Antecedentes de la invención

El diseño moderno de bases de datos comerciales incluye una serie de consideraciones de datos complejos que implican la forma de almacenar, administrar y manipular grandes cantidades de datos. Estos datos incluyen a menudo intrincadas relaciones con otros datos como en un árbol de objetos que proporciona propiedades de herencia entre varios objetos. Estos tipos de relaciones a menudo complican el diseño eficiente de las bases de datos y de los componentes para administrar esos datos. Por ejemplo, uno de los aspectos del procedimiento de diseño de base de datos radica en la comprensión de la forma en que un sistema de gestión de base de datos relacional almacena los datos. Para proporcionar a los usuarios información eficiente y precisa, un programa de base de datos necesita acceder a hechos (datos) sobre diferentes temas almacenados en tablas separadas. Por ejemplo, una tabla solo puede almacenar hechos acerca de los empleados y otra tabla puede almacenar solo hechos con respecto a las ventas, y luego otras tablas para alguna otra materia corporativa. Al utilizar los datos, estos hechos se combinan a continuación y se presentan en muchas formas diferentes de forma automática. Por ejemplo, los usuarios pueden imprimir informes que combinan hechos sobre los empleados y hechos sobre las ventas.

En general, para el diseño de una base de datos, la información se divide en un cierto orden, tales como materias separadas en una biblioteca y luego un programa de base de datos determina cómo se relacionan los sujetos. Estos programas a menudo incluyen una consulta de base de datos relacional utilizando un lenguaje de base de datos común, como Structured Query Language (SQL). Antes de que estos idiomas se puedan aplicar a los datos, varias decisiones se toman generalmente en cuanto a qué tipos de datos son importantes y cómo se deben organizar dichos datos. Por ejemplo, estas decisiones pueden incluir determinar el alcance de una base de datos para decidir cuáles datos almacenar en el mismo. A continuación, la determinación de las tablas necesarias para dividir la información en materias separadas, tales como “empleados” u “órdenes”. Cada tema será entonces una tabla en la base de datos. Otros aspectos incluyen la determinación de los campos respectivos que son necesarios con el fin de decidir qué información almacenar en cada tabla. Cada categoría de información en una tabla se denomina un campo y se muestra como una columna en la tabla. Por ejemplo, un campo en una tabla de empleados podría ser apellidos; otro podría ser fecha de contratación. Otra consideración es determinar las relaciones tales como decidir cómo los datos de una tabla están relacionados con los datos de otras tablas. Los diseñadores suelen añadir campos a las tablas o crear nuevas tablas para clarificar las relaciones, según sea necesario.

Hay varios errores comunes que pueden encontrarse en el diseño de una base de datos. Estos problemas pueden ocasionar que los datos sean más difíciles de usar y mantener. Estos pueden incluir tener una tabla con un gran número de campos en que no todos se refieren al mismo tema. Por ejemplo, una tabla puede contener campos relacionados con los clientes, así como los campos que contienen información de ventas. Además, a menudo es más eficiente si cada tabla contiene datos con respecto a un solo tema. En otros casos, los encabezados se crean cuando los campos se dejan en blanco intencionalmente en muchos registros, ya que no son aplicables a dichos registros. Esto generalmente implica que los campos pertenecen a otra tabla. La redundancia es otro problema cuando un gran número de tablas, muchas de las cuales tienen los mismos campos. Por ejemplo, al separar las tablas de ventas de enero y las ventas de febrero, o para clientes locales y clientes remotos, en los que hay un almacenamiento redundante del mismo tipo de información. Por lo tanto, una técnica está consolidando toda la información relativa a un solo tema en una tabla.

Además de las complejidades de cómo configurar y diseñar las tablas y campos de la base de datos, se deben tomar otras consideraciones. Estos incluyen cómo se debe proporcionar seguridad de datos para las tablas y los campos respectivos (por ejemplo, seguridad, como quién o qué se puede acceder a un archivo). Esto incluye la forma de proporcionar seguridad a las estructuras complejas almacenadas en bases de datos tales como objetos jerárquicos. Clásicamente, las consideraciones de seguridad se han propagado en una jerarquía de herencia de este tipo de objetos, en el que cada elemento de la jerarquía tendría que ser actualizado si uno de los elementos fuera cambiado. Sin embargo, hay un problema común que enfrenta cualquier aplicación que utiliza filas de la tabla de base de datos relacional para almacenar objetos jerárquicos, que es cómo configurar la información de seguridad o datos de cada objeto y rellenar los datos de seguridad para sus objetos secundarios basados en el modelo de herencia. El documento US-B1-6202066 divulga los tipos de acceso de objetos que comprenden las especificaciones de control de acceso que asocian los roles con permisos y asocian estos roles con un conjunto de objetos. El documento da a conocer aún más los tipos de acceso a objetos que son listas de control de acceso NTFS a las especificaciones y los posibles permisos usando los habituales permisos de archivos NTFS.

El documento US 6.105.066 divulga un servidor que incluye una base de datos que almacena los datos de usuario y datos de grupo, como las preferencias del usuario y de grupo y permisos de acceso de usuario del subprograma. El servidor web representa un servidor web típico con soporte para los subprogramas de Java. El administrador de perfil subprogramas del servidor asigna identificaciones de usuarios y grupos a los datos de preferencia. También mantiene una lista de control de acceso para administrar el acceso de usuarios a las aplicaciones en el servidor. Las preferencias de usuario y de grupo se almacenan como una jerarquía de árbol, como se muestra en la figura 3. Todos los usuarios del sistema pertenecen automáticamente al grupo todos los usuarios; este grupo contiene las preferencias por defecto de algunos o todos los subprogramas de usuario en el servidor. Un nuevo grupo tiene acceso a todos los nombres de subprograma permitidos por el propio grupo, así como a los subprogramas permitidos por sus supergrupos. Sin embargo, al igual que Java permite al programador anular un procedimiento de una superclase, la gestión de perfiles permite al administrador del sistema la capacidad de anular un permiso heredado. Esto se llama anular un permiso.

Sumario de la invención

Es el objeto de la presente invención mejorar el manejo de los atributos de seguridad en una base de datos relacional con respecto a los requisitos de procesamiento posteriores a la actualización de los atributos de seguridad. Este objeto se resuelve por la materia objeto de las reivindicaciones independientes. Las realizaciones preferidas se definen en las reivindicaciones dependientes.

A continuación, se presenta un resumen simplificado de la invención con el fin de proporcionar una comprensión básica de algunos aspectos de la invención. Este resumen no es una amplia descripción de la invención. No pretende identificar elementos clave/críticos de la invención o delinear el alcance de la invención. Su único propósito es presentar algunos conceptos de la invención en una forma simplificada como preludio a la descripción más detallada que se presenta más adelante.

La presente invención se refiere a sistemas y procedimientos que proporcionan la seguridad de base regional a una pluralidad de objetos de bases de datos que tienen relaciones jerárquicas entre los objetos. Un componente de región se proporciona que asigna información de seguridad a un subconjunto de los objetos existentes en una jerarquía con el fin de crear una o más zonas de seguridad que son independientes de la jerarquía. Esto permite que los objetos existentes en una región o zona que compartan atributos de seguridad que mitiga los requisitos de procesamiento de la base de datos (por ejemplo, un menor número de nodos en los que actualizar los datos de seguridad). En general, las arquitecturas de bases de datos clásicas a menudo utilizan filas de la tabla de una base de datos relacional para almacenar objetos jerárquicos, que también se provoca que un descriptor de seguridad se ajuste en cada objeto y también rellenar el descriptor de seguridad a los respectivos objetos secundarios basados en el modelo de herencia. Esto provoca cantidades de tiempo de procesamiento cada vez mayores para cada actualización de objeto y se mitiga mediante la introducción de consideraciones de base regional.

Una región puede ser una colección de objetos (no tiene que ser en un árbol contiguo) que comparten la misma o similar descriptor de seguridad. Cuando se actualiza un descriptor de seguridad en un objeto, la región a la que pertenece el objeto puede dividirse o colapsarse. Por ejemplo, una región se puede dividir si un descriptor de seguridad diferente en cualquier objeto hijo es el resultado de la modificación; mientras que una región puede colapsarse en otra región si el cambio resulta en el mismo descriptor de seguridad que el de la otra región. En lugar de que cada objeto posea directamente su propio descriptor de seguridad, una región posee el descriptor de seguridad; lo que reduce drásticamente el número de actualizaciones de objeto cuando se cambia un descriptor de seguridad en un objeto que puede afectar a los descriptores de seguridad de otros objetos.

En general, una región se define clásicamente como un subárbol de objetos en un modelo de objetos jerárquico. En el caso de la presente invención, una región se define como un conjunto de objetos que comparten el mismo descriptor de seguridad, por lo que los objetos que comparten el mismo descriptor de seguridad no tienen que estar bajo el mismo árbol secundario. Este direccionamiento indirecto permite procedimientos eficientes para manipular los descriptores de seguridad de los objetos. Por lo tanto, la seguridad basada en regiones esencialmente transforma un dominio de objeto a un dominio de descriptor de seguridad y lleva a cabo las operaciones de descriptor de seguridad en el dominio de descriptor de seguridad directa e independientemente de la jerarquía que mitiga el procesamiento de base de datos global.

Para la realización de los extremos anteriores y relacionados, se describen ciertos aspectos ilustrativos de la invención en el presente documento en conexión con la siguiente descripción y los dibujos adjuntos. Estos aspectos son indicativos de diversas formas en que la invención puede ponerse en práctica, todos los cuales están destinados a ser cubiertos por la presente invención. Otras ventajas y características novedosas de la invención se harán evidentes de la siguiente descripción detallada de la invención cuando se considera en conjunción con los dibujos.

Breve descripción de los dibujos

La figura 1 es un diagrama de bloques esquemático que ilustra un sistema de seguridad de objeto de acuerdo con un aspecto de la presente invención.

La figura 2 es un diagrama que ilustra un ejemplo de dominio de seguridad transformado de acuerdo con un

aspecto de la presente invención.
La figura 3 ilustra un dominio de seguridad alternativo transformado de acuerdo con un aspecto de la presente invención.
La figura 4 ilustra interfaces de ejemplo de seguridad de acuerdo con un aspecto de la presente invención.
La figura 5 ilustra el procesamiento de componente de región de acuerdo con un aspecto de la presente invención.
La figura 6 ilustra ejemplos de algoritmos de procesamiento de región de acuerdo con un aspecto de la presente invención.
La figura 7 ilustra un procedimiento de región de seguridad de acuerdo con un aspecto de la presente invención.
La figura 8 es un diagrama de bloques esquemático que ilustra un entorno operativo adecuado de acuerdo con un aspecto de la presente invención.
La figura 9 es un diagrama de bloques esquemático de un entorno informático de muestra con el que puede interactuar el objeto de la invención.

Descripción detallada de la invención

La presente invención se refiere a sistemas y procedimientos que proporcionan la seguridad de base regional a los objetos de base de datos que tienen relaciones jerárquicas. En lugar de la actualización de un descriptor de seguridad por separado para cada objeto, la presente invención introduce el concepto de una región, por lo que la seguridad para un objeto dado se deriva de su asociación a la región en oposición a la jerarquía. Esto está en contraste con las arquitecturas clásicas que requieren las descripciones de objetos individuales y tienen la seguridad impuesta por la jerarquía de herencia. De esta manera, el procesamiento y el almacenamiento de bases de datos pueden ser conservados, ya que muchos objetos pueden compartir atributos de seguridad similares que pueden definirse en una escala más global para la región respectiva. En un aspecto, se proporciona un sistema que facilita la gestión de la seguridad y la base de datos. El sistema incluye un componente de base de datos que almacena una pluralidad de objetos que tienen una relación jerárquica entre los objetos. Un componente de región define zonas de seguridad para un subconjunto de objetos y asigna datos de seguridad al subconjunto, en el que las zonas de seguridad son independientes, desacopladas, o disociadas de las relaciones jerárquicas entre los objetos.

Tal como se usa en esta solicitud, los términos “componente”, “sistema”, “objeto”, “zona”, y similares pretenden hacer referencia a una entidad relacionada con los ordenadores, ya sea hardware, una combinación de hardware y software, software, o software en ejecución. Por ejemplo, un componente puede ser, pero no se limita a, un procedimiento que se ejecuta en un procesador, un procesador, un objeto, un ejecutable, un hilo de ejecución, un programa, y / o un ordenador. A modo de ilustración, tanto una aplicación que se ejecuta en un servidor y el servidor pueden ser un componente. Uno o más componentes pueden residir dentro de un procedimiento y/o hilo de ejecución y un componente puede estar localizado en un ordenador y/o distribuido entre dos o más ordenadores. Además, estos componentes pueden ejecutarse desde diversos medios legibles por ordenador que tienen diversas estructuras de datos almacenadas en el mismo. Los componentes pueden comunicarse vía procedimientos locales y/o remotos como de acuerdo con una señal que tiene uno o más paquetes de datos (por ejemplo, los datos de un componente interactuando con otro componente en un sistema local, sistema distribuido, y/o a través de una red como Internet con otros sistemas a través de la señal).

Haciendo referencia inicialmente a la figura 1, un sistema de seguridad de objetos 100 se ilustra de acuerdo con un aspecto de la presente invención. El sistema 100 incluye una base de datos relacional 110 (por ejemplo, SQL u otro tipo de base de datos) que está asociada con un componente de región 120 (o componentes) que define una o más zonas de seguridad del objeto 130. En general, los nodos individuales de una jerarquía de objetos (por ejemplo, ver un objeto de jerarquía en el número de referencia 140) no se actualizan de forma individual cuando se realizan cambios en la seguridad de los objetos. Por el contrario, las políticas de seguridad son asignadas por el componente de región 120 por las respectivas zonas de seguridad en 130. Mediante la asignación de objetos a una zona de seguridad 130 en lugar de actualizar cada objeto individual, el número de operaciones de lectura/escritura en la base de datos 110 puede ser mitigado. Por lo tanto, el componente de región 120 transforma la asignación de política de seguridad de una jerarquía de herencia - donde se actualiza cada objeto, a un dominio de seguridad de objetos en el que las zonas de objetos comparten una política de seguridad similar. De esta manera un subconjunto más pequeño de las actualizaciones de seguridad se puede propagar cuando la política de seguridad de un objeto cambia con solo actualizar el reducido subconjunto de las zonas de seguridad 130 en lugar de actualizar cada objeto individual en una jerarquía de herencia clásica. Se hace notar que los conceptos de herencia se pueden emplear para propagar la política en el sistema 100, sin embargo, la herencia es entre zonas de seguridad 130 en lugar de la herencia convencional entre objetos en un árbol. Por lo tanto, la herencia se produce entre los componentes que se modelan en un dominio de seguridad en lugar de un dominio de objeto. Esto implica que las asignaciones de seguridad para un objeto en cuestión son entre el objeto y su zona asociada 130 más que explícitamente establecidas para el objeto individual 140. Por lo tanto, el componente de región 120 proporciona seguridad a una región de objetos identificados y esencialmente se desacopla, se disocia, o es independiente de las jerarquías de objetos convencionales que propagan los cambios de seguridad a todos los objetos de la jerarquía.

En general, a los elementos de la base de datos 110 se puede asignar un ID (identificador) para un descriptor de seguridad. La base de datos incluye una tabla [Table!Item] que tiene una columna llamada SDID (ID descriptor de seguridad). Este SDID es un identificador único de un descriptor de seguridad que se almacena y se mantiene en

una tabla del sistema de SQL oculta, por ejemplo. Una tabla de sistema puede estar expuesta a través de una vista pública (por ejemplo, Sys. Descriptor de Seguridad). La siguiente tabla es una ilustración simplificada de cómo un descriptor de seguridad puede ser conectado o asociado con un modelo de objeto básico:

[Table!Item]: Asocia un elemento con un ID descriptor de seguridad.

_ID del objeto	...	_SDID	…

[Sys. Descriptor de seguridad]: Asigna el ID para el contenido del descriptor de seguridad.

Sd_id	Tipo	Descriptor de seguridad	…

Para asignar de manera eficiente un ID de descriptor de seguridad (SD ID) a un elemento de objeto, una tecnología de región SD se basa en parte en la observación de que la mayoría de los elementos objeto tienden a compartir el mismo descriptor de seguridad. Una región SD es un conjunto de elementos (que no tienen que ser contiguos como en los sistemas convencionales) que comparten la misma o similar ID SD. Por lo general, todos los elementos de la [Table!Item] que se muestra más arriba se pueden agrupar en diferentes regiones SD. La relación de las regiones SD puede establecerse de tal manera que la desviación estándar de una región SD puede heredar del SD de otra región SD en el dominio de seguridad que se ha descrito anteriormente. En esencia, se establece un árbol de región SD que es comparable con el árbol de elementos de objeto correspondiente, pero con un menor número de nodos como se mostrará con respecto a las figuras 2 y 3 a continuación. El árbol de región SD por lo tanto se puede utilizar para actualizar el SD del elemento de manera eficiente. Normalmente cuando se crea un árbol de elementos de seguridad, se crean tres regiones SD para asignar los SDs sustancialmente a todos los elementos en el árbol. Por lo tanto, una región SD es para el elemento raíz (donde se define un SD explícita), otra región SD es para los respectivos elementos de contenedores y el último SD para elementos no contenedores.

Con referencia ahora a las figuras 2 y 3, el dominio de seguridad de ejemplo transformados 200 y 300 se ilustran de acuerdo con un aspecto de la presente invención. En 200 de la figura 2, se ilustran los nodos de un árbol de objeto, en donde un nodo negro en 210 es un elemento de la raíz; los nodos grises en 220 son elementos contenedores, y los nodos blancos en 230 son elementos no contenedores. Cuando se actualiza el descriptor de seguridad de un elemento (SD) (por ejemplo, cambiando el propietario del SD, grupo, lista de control de acceso y así sucesivamente), la región SD donde pertenece el elemento se puede dividir en tres subgrupos o subconjuntos, como se ilustra en 240. Los cambios en la seguridad generalmente se efectúan a través de los datos referidos como una entrada de control de acceso (ACE) que puede ser de una forma explícita o implícita. Cuando las ACE explícitas se añaden al SD de un elemento, nuevas regiones SD se pueden crear en torno a este tema. En este caso, se crean tres regiones SD, una para el elemento (donde se añaden las ACE explícitas) en sí, uno para sus hijos contenedores y una para sus hijos no contenedores. Haciendo referencia a la figura 3, una situación más compleja se ilustra cuando se agrega una ACE explícita no propagada a un SD en un elemento en 310, en cuyo caso cinco nuevas regiones se crean en torno al elemento, como se ilustra en 320. En este caso, se crea una región para el elemento en sí mismo (donde se añade la ACE explícita) 330, una región para sus hijos contenedores directos en, una región para sus hijos directos no contenedores en 350, una región para sus hijos no directos contenedores en 360 y una región para sus hijos no contenedores no directos en 370.

Para resumir las figuras 2 y 3, las nuevas regiones se pueden crear cuando el SD de un elemento se actualiza de forma explícita (no a través de la herencia). Generalmente, 3 o 5 nuevas regiones (posibles otras cantidades) se crean en función de los cambios que se realizan para la SD. Cinco regiones SD se crean si se añade una ACE no propagada y tres regiones SD se crean generalmente en otros casos. A modo de ejemplo, supongamos que el elemento cuyo SD contiene propiedades no heredadas (ACE no hereditarias en la mayoría de los casos) como el elemento raíz. Como se señaló anteriormente, un elemento raíz de tipo contenedor puede ser dueño de 3 o 5 regiones SD dependiendo de los tipos de ACE explícitas en el SD. Un no contenedor puede tener su propia región SD si su SD tiene propiedades explícitas. Si todas las propiedades explícitas de SD de un elemento raíz se eliminan, a continuación, las regiones SD poseídas por este elemento raíz pueden ser colapsadas en el SD de su elemento padre que a su vez reduce las actualizaciones de seguridad de objetos individuales. Cada región SD se puede representar como una fila en una tabla Security_Hierachy como el siguiente ejemplo:

[Table!Security_Hierachy]: Almacena la relación de herencia SD y establece los elementos a compartir el mismo descriptor de seguridad.

_SDIdParent	_SDId	_RootItemID	_IsContainer	_Scope

Las columnas de la tabla anterior pueden incluir un _SDId que es el ID de la región SD, un campo _SDIdParent que es el ID del SD en las propiedades de seguridad heredadas son procedentes de, un campo _RootItemID que es el ID del elemento en el que el SD explícito es definido, un campo _IsContainer que es 1 si el SD se aplica al contenedor, o 0 a un no contenedor, y un campo de _Alcance que se codifica como sigue:

0: el SD se aplica al Elemento Raíz. 1: el SD solo se aplica a hijos del elemento Raíz. 2: el SD se aplica a los hijos directos del Elemento Raíz. 3: el SD se aplica a los hijos no directos del Elemento Raíz.

Se observa que cuando una base de datos se automantiene, tres descriptores de seguridad predeterminados se pueden crear si se desea; un descriptor de la parte superior del elemento Raíz, un descriptor para todos los hijos contenedores y un descriptor para todos los hijos que no son de contenedores. En consecuencia, tres regiones SD en la parte superior del elemento Raíz se pueden crear también. Por lo general, todos los elementos creados posteriormente en el volumen pueden tener uno de los SDs como su SD predeterminado. Cuando las ACEs explícitas se añaden al elemento, se pueden crear nuevas regiones SD como se discutió anteriormente.

La figura 4 ilustra interfaces de seguridad de ejemplo 400 de acuerdo con un aspecto de la presente invención. Varias interfaces de seguridad 400 se pueden proporcionar para interactuar con las consideraciones basadas en regiones descritas anteriormente. A continuación, se describirán solo algunos ejemplos de interfaz que pueden ser aplicados. Estos pueden incluir interfaces para la recuperación de datos de seguridad en 410, interfaces para configurar la información de seguridad en 420, e interfaces para mantener enlaces, como se describirá en más detalle a continuación. El fragmento de código siguiente es un ejemplo de una declaración pública para algunas de estas interfaces 400.

Elemento de seguridad de clase sellado público

```
{
      public ItemSecurity( Guid itemId )
      public string GetSDDLSecurity()
      public GenericSecurityDescriptor GetSecurity()
      public void SetSDDLSecurity( string sd, SECURITY_INFORMATION
      si )
      public void SetSecurity( GenericSecurityDescriptor gsd ,
                                    SECURITY_INFORMATION si)
      public string GetUserEffectiveSecurity()
      public void AddHoldingLink( Guid itemId )
      public void RemoveHoldingLink( Guid itemId )
}
```

A continuación, se ofrece una breve descripción de las interfaces de seguridad 410 a 430:

public string GetSDDLSecurity() - Recupera todo el descriptor de seguridad en el elemento en formato de cadena SDDL. Incluye listas de control de acceso heredadas y explícitas.
public GenericSecurityDescriptor GetSecurity() - Recupera todo el descriptor de seguridad en el elemento en el formato de una clase Gestionado GenericSecurityDescriptor ACL.
SetSDDLSecurity public void (sd cadena, SEGURIDAD DE LA INFORMACIÓN SI) Establece el descriptor de seguridad en el elemento. Esta función hace caso omiso de las ACE heredadas. Se vuelve a generar las ACE heredadas de su padre y otros enlaces que llevan a cabo. Se le puede llamar para establecer el propietario, grupo, indicador de control o las ACE explícitas. SECURITY_INFORMATION especifica qué parte del descriptor de seguridad se va a actualizar.
pública SetSecurity vacío (GenericSecurityDescriptor GSD, si SECURITY_INFORMATION) - Establece el descriptor de seguridad en el elemento. Toma la clase ACL Gestionado como parámetro de entrada.
AddHoldingLink public void (Guid itemId) - Actualiza el descriptor de seguridad en el elemento cuando se añade un nuevo enlace de sujeción al elemento.
RemoveHoldingLink public void (Guid itemId) - Actualiza el descriptor de seguridad en el elemento al retirar un nuevo enlace de sujeción del elemento.
GetUserEffectiveSecurity cadena pública () - Recuperar el descriptor de seguridad en el elemento que contiene

las entradas de control pertinentes al contexto de seguridad actual.

La figura 5 ilustra un procesamiento de componente de región 500 de acuerdo con un aspecto de la presente invención. En 510, se proporcionan definiciones de región. Estas incluyen una región del descriptor de seguridad (SD), que es un conjunto de objetos que comparten el mismo SD. El conjunto de elementos no tiene que formar un árbol contiguo. Una fila de jerarquía de seguridad (SH) es una fila de una [Tabla!Security_Hierachy] de la siguiente lista. Cada región SD debe tener una fila de SH en la tabla.

_ParentSDId	_SDId	_RootItemId	_IsContainer	_Scope
SD0	SD1	ItemId	0	3

Una fila en la tabla anterior se refiere como una fila de SH que corresponde a una región de SD. Las filas de esta tabla indican un conjunto de elementos (puede ser un solo elemento) que comparten el mismo descriptor de seguridad (SD1 en el ejemplo anterior). El conjunto de elementos se define por una raíz común (el ItemId), un tipo común (contenedor o no contenedor) y un ámbito. El ámbito es opcional para soportar diferentes modelos de seguridad del sistema operativo.

En 520, se describen consideraciones de combinación de región y de creación. En este aspecto, una nueva región SD puede crearse bajo las siguientes condiciones:

1. Cambios SD realizados en un elemento no contenedor.

Tres nuevas regiones SD pueden crearse en las siguientes condiciones:

1. Cambios SD realizados en un elemento contenedor, y

2. Los cambios SD no incluye las ACEs sin propagación.

Cinco nuevas regiones SD podrían crear en las siguientes condiciones:

1. Cambios SD realizados en un elemento contenedor, y

2. Los cambios SD incluyen ACEs sin propagación.

Las regiones SD pueden fusionarse bajo las siguientes condiciones:

1. SD padre hace cumplir la herencia SD mediante el lavado de los SDs hijo; o

2. Las ACEs explícitas se eliminan de un SD.

En 530, se proporcionan diversas nociones que se puede emplear en los siguientes algoritmos descritos con respecto a la figura 6. Estas anotaciones incluyen:

_Item o * - El sistema del elemento actual que aplica las operaciones on.
SDId (x) o SDId - El sd_id del descriptor de seguridad sobre el punto x.
SDId_NC (x) o SDId_NC - El SDId se aplica a los objetos hijo sin contenedores del elemento x.
SDId_C (x) o SDId_C - El SDId se aplica a los objetos hijo de contenedor del elemento x.
SDId_NC2 (x) o SDId_NC2 - El SDId se aplica para dirigir los objetos hijo sin contenedores del elemento x.
SDId_C2 (x) o SDId_C2 - El SDId se aplica para dirigir los objetos hijo de contenedor del elemento x.
SDId_NC3 (x) o SDId_NC3 - El SDId se aplica a los objetos hijo sin contenedor no directos del elemento x.
SDId_C3 (x) o SDId_C3 - El SDId se aplica a los objetos hijo de contenedor no directos del elemento x.
SHRow (x, i, j) - La fila en la tabla [Table!Security_Hierachy] donde _RootItemId = x, _IsContainer = i, j = _Scope
UpdateIternSD (OldSDId, NewSDId, RootItem, IsContainer, Scope) - Modificar el SDID de todos los elementos del tipo (IsContainer) cuyo SDiD actual = OldSDId, el antepasado es RootItem dentro del Scope a NewSDId.

UpdateSDBlob (SDiD) - Actualiza el contenido de los descriptores de seguridad de este SDID y sus hijos si el SDID de sus hijos no forma un ciclo con este SDiD. Por ejemplo, cuando se añade un enlace de retención (con SD0) a un elemento de archivo (con SD1) que no tiene su propia fila en la tabla [Table!Security_Hierachy], se crearán tres filas (SD0, SD1, _Item, 0, 0), (SD1, SD0, _Item, 0, 1), (SD1, SD0, _Item, 1,1). Aquí se reutiliza SD0 para los elementos hijo de este elemento para reducir significativamente el número de cambios en la tabla [Table!Item].

UpdateSDId (SDiD, SDId_New) - Actualiza las filas del elemento actual de [Table!Security_Hierachy] donde _SDId = SDID para establecer SDiD = SDId_New.

UpdateParentSDId (SDIdPar, SDIdPar_New) - Actualiza las filas de [Table!Security_Hierachy] donde _ParentSDId =

SDIdPar para establecer _ParentSDId = SDIdPar_New.

CreateNewSD (SDiD) - Crea un nuevo SD del SD actual más los cambios que se realicen (añadir/quitar las ACEs, añadir/eliminar enlaces de retención).

La figura 6 ilustra algoritmos de procesamiento 600 de región de ejemplo de acuerdo con un aspecto de la presente invención. En este aspecto, al menos tres algoritmos 600 separados o combinados se puede emplear para efectuar procedimientos de región. Estos incluyen un Ajuste de descriptor de seguridad en 610; un Añadir enlace de retención 620; y un Eliminar algoritmo de enlace de retención en 630. Con respecto al Ajuste de descriptor de seguridad 610, hay varias maneras de cambiar el descriptor de seguridad en un objeto que al menos incluya:

- Añadir/Eliminar ACEs explícitas no hereditarias.
- Añadir/Eliminar ACEs explícitas hereditarias que se aplican a este elemento y a todos sus hijos.
- Añadir/Eliminar ACEs explícitas hereditarias que solo se aplican a sus hijos.
- Añadir/Eliminar ACEs explícitas hereditarias que solo se aplican a este elemento y a sus hijos directos.
- Añadir/Eliminar ACEs explícitas hereditarias que solo se aplican a los contenedores hijo.
- Añadir/Eliminar ACEs explícitas hereditarias que solo se aplican a los objetos hijo.
- Añadir/Eliminar ACEs explícitas hereditarias que solo se aplican a cierto tipo de objetos.
- Cambiar el propietario del descriptor de seguridad.
- Cambiar el grupo de descriptor de seguridad.
- Cambiar los indicadores de control del descriptor de seguridad.

  i. Detener la herencia de ACEs
  ii. Iniciar la herencia de ACEs
  iii. Cambiar otros indicadores de control que solo se aplican a este elemento.

En 620, cuando se añade un enlace de retención a un elemento, el descriptor de seguridad sobre este elemento puede cambiarse o no, dependiendo de si el enlace de retención dispone de ACEs hereditarias y si el SD en este elemento tiene un indicador SE_DACLE_PROTECTED activado. Sin embargo, la tabla [Table!Security_Hierachy] debe ser actualizada. Cuando se añade un enlace de retención a un elemento, tres nuevas filas para el elemento deben añadirse a la tabla [Table!Security_Hierachy] si el elemento no tiene una fila designada todavía. Para reducir la actualización en la tabla [Table!Item], los siguientes formatos se pueden utilizar para crear estas filas: (SD0, SD1, *, 0, 0), (SD1, SD0, *, 0, 1), (SD1, SD0, *, 1, 1) donde SD0 es el antiguo SDID del elemento de destino del enlace de retención, SD1 es el nuevo SDID del elemento de destino. Mediante este esquema, solo tendrá que actualizar el elemento de fuente en la tabla [Table!Item]. Sobre la base de este esquema, si se agrega una ACE hereditaria no explícita sobre este dato más adelante, no se realiza una actualización en la tabla [Table!Item]. En 630, se puede suponer que la SDID del descriptor de seguridad en el enlace de retención que ser eliminado es SDId_HD. En el caso de la eliminación del enlace de retención, las regiones SD se pueden colapsar y, por lo tanto, las filas de la tabla [Table!Security_Hierachy] se pueden intercalar.

La figura 7 ilustra un procedimiento 700 de región de seguridad de ejemplo para la seguridad de objetos de base de datos de acuerdo con un aspecto de la presente invención. Aunque, a los efectos de simplificar la explicación, la metodología se muestra y se describe como una serie o número de actos, se debe entender y apreciar que la presente invención no está limitada por el orden de los actos, ya que algunos actos pueden, de acuerdo con la presente invención, producirse en diferentes órdenes y/o al mismo tiempo con otros actos de los que se muestran y describen en este documento. Por ejemplo, los expertos en la técnica entenderán y apreciarán que una metodología podría alternativamente representarse como una serie de estados o eventos interrelacionados, como en un diagrama de estado. Por otra parte, no todos los actos ilustrados pueden ser requeridos para implementar una metodología de acuerdo con la presente invención.

Pasando a 710 de la figura 7, los descriptores de seguridad para los objetos respectivos en una base de datos están desacoplados o disociados de una jerarquía de objetos clásica mediante la eliminación del requisito de que cada objeto que se actualiza (en cuanto a seguridad) a la vista de cualquier posible actualización de la jerarquía. En 720, uno o más descriptores de seguridad se utilizan para definir las regiones de objeto para los objetos que residen en la base de datos. Como se señaló anteriormente, esto puede incluir el colapso o la fusión de los datos de seguridad objeto de árboles de objetos similares o diferentes con el fin de definir las regiones de seguridad o subconjuntos de objetos que se suscriben a los datos de seguridad similares de la región. Además, tales datos de región se pueden definir en una fila de la base de datos incluyendo relaciones resultantes con otros objetos que pertenecen a la región. En 730, las políticas de seguridad de los objetos se establecen por regiones seleccionadas en la base de datos. Como se señaló anteriormente, dependiendo del tipo de entrada de control de acceso (Implícito/Explícito) y la ubicación de un cambio de seguridad en una jerarquía de objetos, se pueden crear varias regiones de seguridad a partir de tales ajustes. En 740, se producen transformaciones entre dominios de objetos clásicos y los dominios de seguridad de la presente invención con el fin de propagar los cambios de seguridad dentro de la base de datos. Esto puede incluir la creación de subconjuntos de regiones alrededor de un objeto dado en el momento de que se solicite un cambio de seguridad para el objeto (por ejemplo, crear tres o cinco regiones en función del tipo de cambio de seguridad).

Con referencia a la figura 8, un entorno 810 de ejemplo para implementar diversos aspectos de la invención incluye un ordenador 812. El ordenador 812 incluye una unidad de procesamiento 814, una memoria de sistema 816, y un bus de sistema 818. El bus del sistema 818 acopla los componentes del sistema incluyendo, pero no limitado a, la memoria del sistema 816 a la unidad de procesamiento 814. La unidad de procesamiento 814 puede ser cualquiera de los diversos procesadores disponibles. Microprocesadores duales y otras arquitecturas de multiprocesadores también se pueden emplear como la unidad de procesamiento 814.

El bus del sistema 818 puede ser cualquiera de varios tipos de estructura(s) de bus, incluyendo el bus de memoria o el controlador de memoria, un bus periférico o un bus externo, y/o un bus local usando cualquier variedad de arquitecturas de bus disponibles, incluyendo, pero no limitado a, bus de 11 bits, Arquitectura Estándar Industrial (ISA), Arquitectura de microcanal (MSA), ISA extendida (EISA), Electrónica de accionamiento inteligente (IDE), bus local VESA (VLB), interconexión de componentes periféricos (PCI), bus serie universal (USB), puerto de gráficos avanzado (AGP), bus de asociación internacional de tarjetas de memoria de ordenador personal (PCMCIA), e interfaz de pequeños sistemas de ordenador (SCSI).

La memoria del sistema 816 incluye una memoria volátil 820 y una memoria no volátil 822. El sistema básico de entrada/salida (BIOS), que contiene las rutinas básicas para transferir información entre elementos dentro del ordenador 812, como durante el arranque, se almacena en la memoria no volátil 822. A modo de ilustración, y no de limitación, la memoria no volátil 822 puede incluir una memoria de solo lectura (ROM), ROM programable (PROM), ROM eléctricamente programable (EPROM), ROM borrable eléctricamente (EEPROM), o memoria flash. La memoria volátil 820 incluye una memoria de acceso aleatorio (RAM), que actúa como una memoria caché externa. A modo de ilustración y no de limitación, la RAM está disponible en muchas formas, tales como RAM síncrona (SRAM), RAM dinámica (DRAM), DRAM síncrona (SDRAM), SDRAM de doble velocidad de datos (DDR SDRAM), SDRAM aumentada (ESDRAM), DRAM SynchLink (SLDRAM), y RAM Rambus directa (DRRAM).

El ordenador 812 incluye además medios extraíbles/no extraíbles volátiles/no volátiles de almacenamiento informático. La figura 8 ilustra, por ejemplo, un almacenamiento en disco 824. El almacenamiento en disco 824 incluye, pero no se limita a, dispositivos como una unidad de disco magnético, unidad de disquete, unidad de cinta, unidad Jaz, unidad Zip, unidad LS-100, tarjeta de memoria flash, o lápiz de memoria. Además, el almacenamiento en disco 824 puede incluir medios de almacenamiento por separado o en combinación con otros medios de almacenamiento, incluyendo, pero no limitado a, una unidad de disco óptico tal como un dispositivo de disco compacto ROM (CD-ROM), unidad CD grabable (unidad CD-R), CD regrabable (unidad CD-RW) o una unidad de disco digital versátil ROM (DVD-ROM). Para facilitar la conexión de los dispositivos de almacenamiento en disco 824 al bus de sistema 818, se suele utilizar una interfaz extraíble o no extraíble como interfaz 826.

Debe apreciarse que la figura 8 describe software que actúa como un intermediario entre los usuarios y los recursos informáticos básicos descritos en el entorno operativo 810 adecuado. Este tipo de software incluye un sistema operativo 828. El sistema operativo 828, que puede almacenarse en un almacenamiento de disco 824, que actúa para controlar y asignar recursos del sistema de ordenador 812. Las aplicaciones del sistema 830 se aprovechan de la gestión de los recursos mediante el sistema 828 que opera a través de unos módulos de programa 832 y 834 del programa los datos almacenados en la memoria del sistema 816 o en el almacenamiento en disco 824. Debe apreciarse que la presente invención puede implementarse con diversos sistemas operativos o combinaciones de sistemas operativos.

Un usuario introduce comandos o información en el ordenador 812 a través del dispositivo(s) de entrada 836. Los dispositivos de entrada 836 incluyen, pero no se limitan a, un dispositivo señalador tal como un ratón, bola de seguimiento, lápiz óptico, pantalla táctil, teclado, micrófono, joystick, almohadilla de juegos, antena parabólica, escáner, tarjeta sintonizadora de TV, cámara digital, cámara de vídeo digital, cámara web, y similares. Estos y otros dispositivos de entrada se conectan a la unidad de procesamiento 814 a través del bus del sistema 818 a través de puerto(s) de interfaz 838. El(los) puerto(s) de interfaz 838 incluye(n), por ejemplo, un puerto serie, un puerto paralelo, un puerto de juegos, y un bus serie universal (USB). El(los) dispositivo(s) de salida 840 utiliza(n) alguna del mismo tipo de los puertos como dispositivo(s) de entrada 836. Así, por ejemplo, un puerto USB puede utilizarse para proporcionar la entrada al ordenador 812, y la información de salida desde el ordenador 812 a un dispositivo de salida 840. Un adaptador de salida 842 se proporciona para ilustrar que hay algunos dispositivos de salida 840 como monitores, altavoces e impresoras, entre otros dispositivos de salida 840, que requieren adaptadores especiales. Los adaptadores de salida 842 incluyen, a modo de ilustración y no de limitación, tarjetas de vídeo y de sonido que proporcionan un medio de conexión entre el dispositivo de salida 840 y el bus 818 del sistema. Debe tenerse en cuenta que otros dispositivos y/o sistemas de dispositivos proporcionan ambas capacidades de entrada y salida, como equipo(s) remoto(s) 844.

El ordenador 812 puede operar en un entorno de red usando conexiones lógicas a uno o más ordenadores remotos, tal como ordenador(es) remoto(s) 844. El ordenador(es) remoto(s) 844 puede(n) ser un ordenador personal, un servidor, un enrutador, un PC de red, una estación de trabajo, un aparato basado en un microprocesador, un dispositivo par u otro nodo de red común y similares, y típicamente incluye muchos o todos los elementos descritos con relación al ordenador 812. Para fines de brevedad, solamente un dispositivo de almacenamiento de memoria 846 se ilustra con el ordenador(es) remoto(s) 844. El ordenador(es) remoto(s) 844 está(n) conectado(s) lógicamente al ordenador 812 a través de una interfaz de red 848 y luego físicamente conectado a través de conexión de

comunicación 850. La interfaz de red 848 abarca las redes de comunicación, como las redes de área local (LAN) y redes de área amplia (WAN). Las tecnologías LAN incluyen interfaz de datos distribuidos de fibra (FDDI), interfaz de datos distribuidos de cobre (CDDI), Ethernet/IEEE 802.3, Token Ring/IEEE 802.5 y similares. Las tecnologías WAN incluyen, pero no se limitan a, enlaces punto a punto, redes de conmutación de circuitos como servicios integrados de redes digitales (RDSI) y variaciones de los mismos, redes de conmutación de paquetes, y líneas de abonado digital (DSL).

Conexión(es) de comunicación 850 se refiere(n) al hardware/software empleado para conectar la interfaz de red 848 al bus 818. Aunque se muestra la conexión de comunicación 850 para mayor claridad ilustrativa dentro del ordenador 812, también puede ser externa al ordenador 812. El hardware/software necesario para la conexión a la interfaz de red 848 incluye, a modo de ejemplo solamente, tecnologías internas y externas tales como módems, incluyendo módems telefónicos de grado regular, módems de cable y módems DSL, adaptadores ISDN, y tarjetas Ethernet.

La figura 9 es un diagrama de bloques esquemático de un entorno de computación de muestra 900 con los que puede interactuar el objeto de la invención. El sistema 900 incluye uno o más clientes 910. El cliente(s) 910 puede ser hardware y/o software (por ejemplo, hilos, procedimientos, dispositivos de computación). El sistema 900 también incluye uno o más servidores 930. El servidor(es) 930 también puede(n) ser hardware y/o software (por ejemplo, hilos, procedimientos, dispositivos de computación). Los servidores 930 pueden albergar hilos para realizar transformaciones mediante el empleo de la presente invención, por ejemplo. Una posible comunicación entre un cliente 910 y un servidor 930 puede ser en forma de un paquete de datos adaptado para ser transmitido entre dos o más procedimientos informáticos. El sistema 900 incluye un marco de comunicación 950 que se puede emplear para facilitar las comunicaciones entre el cliente(s) 910 y el servidor(es) 930. El cliente(s) 910 está(n) conectado(s) operativamente a uno o más almacenes de datos de cliente 960 que pueden emplearse para almacenar la información local para el cliente(s) 910. Del mismo modo, el servidor(es) 930 está(n) conectado(s) operativamente a uno o más almacenes de datos del servidor 940 que puede emplearse para almacenar información local a los servidores 930.

Lo que se ha descrito anteriormente incluye ejemplos de la presente invención. Por supuesto, no es posible describir cada combinación concebible de componentes o metodologías a efectos de describir la presente invención, pero un experto normal en la técnica puede reconocer que muchas otras combinaciones y permutaciones de la presente invención son posibles. En consecuencia, la presente invención pretende abarcar todas las alteraciones, modificaciones y variaciones que caen dentro del alcance de las reivindicaciones adjuntas. Además, en la medida en que se utiliza el término “incluye”, ya sea en la descripción detallada o en las reivindicaciones, tal término pretende ser inclusivo, de una manera similar a la expresión “que comprende”, porque “que comprende” se interpreta cuando se emplea como una palabra transitoria en una reivindicación.

REIVINDICACIONES

1. Un sistema (100) que facilita seguridad y gestión de bases de datos, incluyendo el sistema:

una base de datos relacional (110) adaptada para almacenar una pluralidad de objetos que tienen una relación jerárquica entre los objetos, comprendiendo esos objetos un objeto de raíz, objetos hijos contenedores, y objetos hijos no contenedores; y
un componente de región (120) asociado a la base de datos relacional adaptado para establecer un árbol de descriptor de seguridad (SD), comprendiendo dicho árbol SD regiones SD para el objeto de raíz, los objetos de contenedor, y dichos objetos no contenedores, y adaptado para la transformación de política de seguridad de asignación de una jerarquía de herencia a un dominio de seguridad de objetos, donde las zonas de seguridad de objetos comparten una política de seguridad similar,

dicho componente de región (120) adaptado para definir por medio de dichas regiones SD una o más zonas de seguridad (130) para un subconjunto de los objetos, en el que cada zona de seguridad por lo tanto tiene un descriptor de seguridad (SD) asociado, siendo las regiones SD independientes de las relaciones jerárquicas entre los objetos, y en el que el componente de región está adaptado además para asignar una de las una o más regiones SD a cada uno de la pluralidad de objetos, de tal manera que los objetos se agrupan en base a la región SD a las que se asignan los objetos, con lo que permite el cambio de un descriptor de seguridad de un objeto mediante la modificación de la región SD a la que está asignado el objeto; y
adaptado para crear una nueva primera región SD (330) para un objeto (310), si el descriptor de seguridad de dicho objeto (310) se actualiza y para crear una nueva segunda región SD (340) asignada a los hijos de contenedores de dicho objeto actualizado y una nueva tercera región SD (350) asignada a los hijos no contenedores de ese objeto actualizado.

2. El sistema de la reivindicación 1, en el que el componente de región (120) soporta seguridad de herencia entre regiones SD en el dominio de seguridad en el que una región SD se define como un conjunto de objetos que tienen el mismo descriptor de seguridad.

3. El sistema de la reivindicación 1, en el que el componente de región (120) soporta un colapso de las regiones SD basadas en un análisis de los cambios de seguridad.

4. El sistema de la reivindicación 3, en el que el componente de región (120) expande las regiones SD mediante al menos tres regiones SD basadas en los cambios de seguridad detectados.

5. El sistema de la reivindicación 4, en el que los cambios de seguridad son detectados por una entrada de control de acceso, ACE.

6. El sistema de la reivindicación 5, en el que la entrada de control de acceso representa un cambio de seguridad explícito o implícito.

7. El sistema de la reivindicación 1, que comprende además una tabla que asocia un elemento de objeto con un identificador de descriptor de seguridad.

8. El sistema de la reivindicación 7, que comprende además una tabla que asigna el identificador de descriptor de seguridad a contenidos de un descriptor de seguridad.

9. El sistema de la reivindicación 1, que comprende además al menos una interfaz para interactuar con el componente de región (120) o la base de datos relacional (110).

10. El sistema de la reivindicación 9, en el que la interfaz incluye una función de obtención de seguridad, unas funciones de obtención de descriptores, una función de ajuste de seguridad, una función de añadir un enlace de retención, una función de retirar un enlace de retención, y una función de obtención de una seguridad efectiva.

11. El sistema de la reivindicación 1, que comprende además una fila de jerarquía de seguridad para definir las relaciones de objetos de seguridad en un dominio de seguridad.

12. Un procedimiento para facilitar seguridad y gestión de bases de datos, comprendiendo el procedimiento las etapas de:

definir objetos de la base de datos en un dominio de objetos de una base de datos relacional, teniendo los objetos de la base de datos una relación jerárquica entre los objetos, comprendiendo esos objetos un objeto de raíz, objetos hijos contenedores, y los objetos hijos no contenedores;
establecer por medio de un componente de región (120) un árbol de descriptor de seguridad (SD), comprendiendo dicho árbol SD regiones SD para el objeto de raíz, los objetos contenedores, y los objetos no contenedores;
transformar la asignación de política de seguridad de una jerarquía de herencia a un dominio de seguridad de objetos, donde las zonas de seguridad de los objetos comparten una política de seguridad similar;

definir mediante el componente de región (120) por medio de dichas zonas de seguridad de las regiones SD en un dominio de seguridad para un subconjunto de los objetos de la base de datos, en el que cada zona de seguridad por lo tanto tiene un descriptor de seguridad (SD) asociado, siendo las regiones SD independientes de las relaciones jerárquicas entre los objetos de la base de datos;

asignar mediante el componente de región (120) una de las una o más regiones SD a cada uno de la pluralidad de objetos de tal manera que los objetos se agrupan en base a la región SD a los que se asignan los objetos, lo que permite el cambio del descriptor de seguridad de un objeto mediante la modificación de la región SD a la que está asignada el objeto; y

crear mediante el componente de región (120) una nueva primera región SD (330) para un objeto (310), si un descriptor de seguridad de dicho objeto (310) se actualiza y crear una nueva segunda región SD (340) asignada a los hijos contenedores de dicho objeto actualizado y una nueva tercera región SD (350) asignada a los hijos no contenedores de ese objeto actualizado.

13. El procedimiento de la reivindicación 12, que comprende además la generación de al menos tres regiones SD tras detectar cambios en el dominio de seguridad.

14. El procedimiento de la reivindicación 12, que comprende además proporcionar un mecanismo de herencia para las regiones SD dentro del dominio de seguridad.

15. Un medio legible por ordenador que tiene instrucciones legibles por ordenador almacenadas en el mismo para implementar el procedimiento de la reivindicación 12.

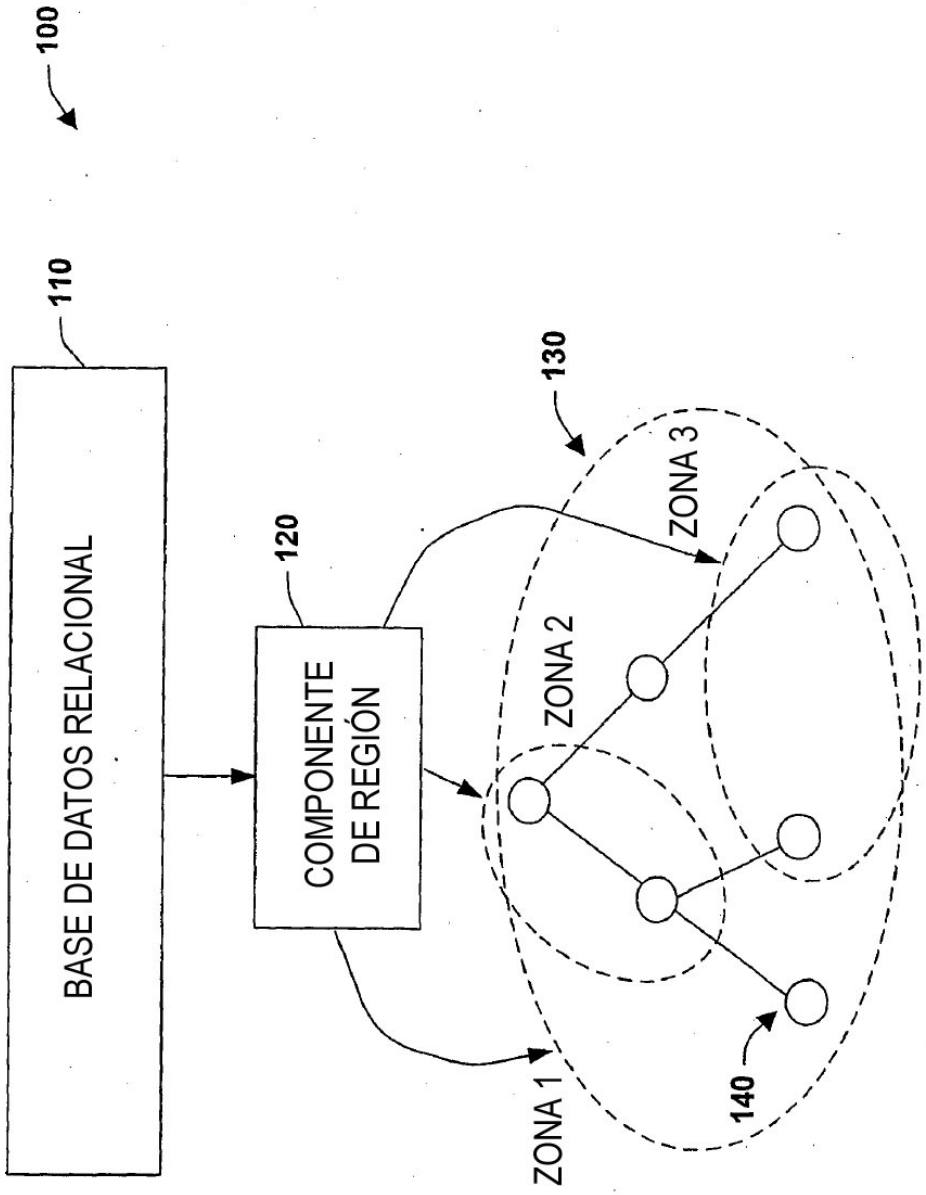
100
110
BASE DE DATOS RELACIONAL
120
COMPONENTE DE REGIÓN
130
ZONA 1
ZONA 2
ZONA 3
ZONA N
140

FIG. 1

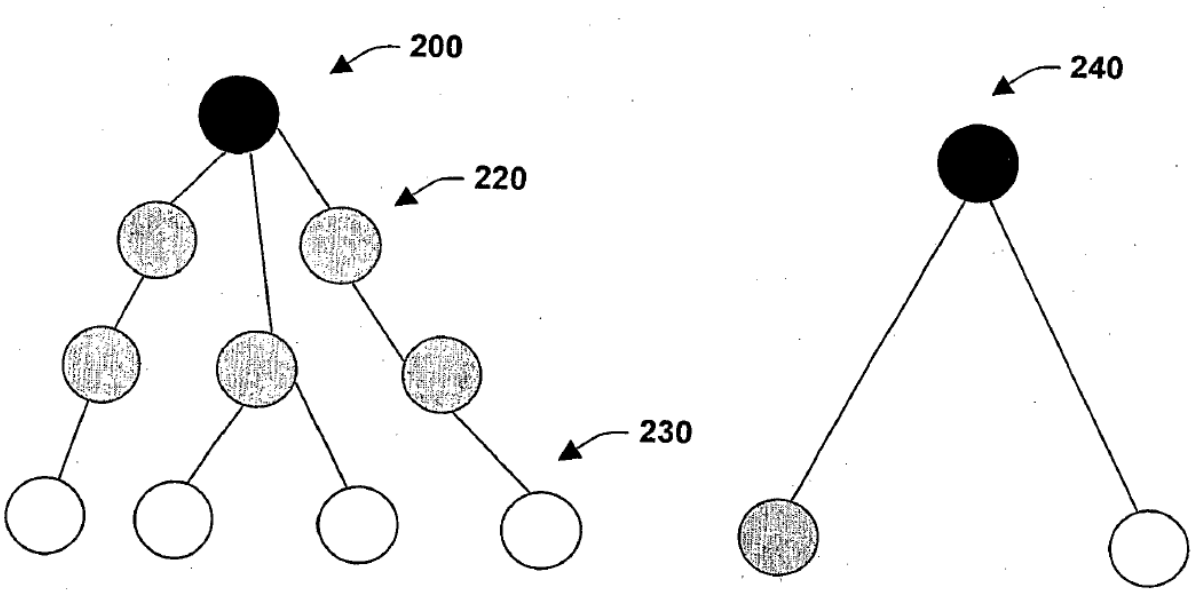

FIG. 2

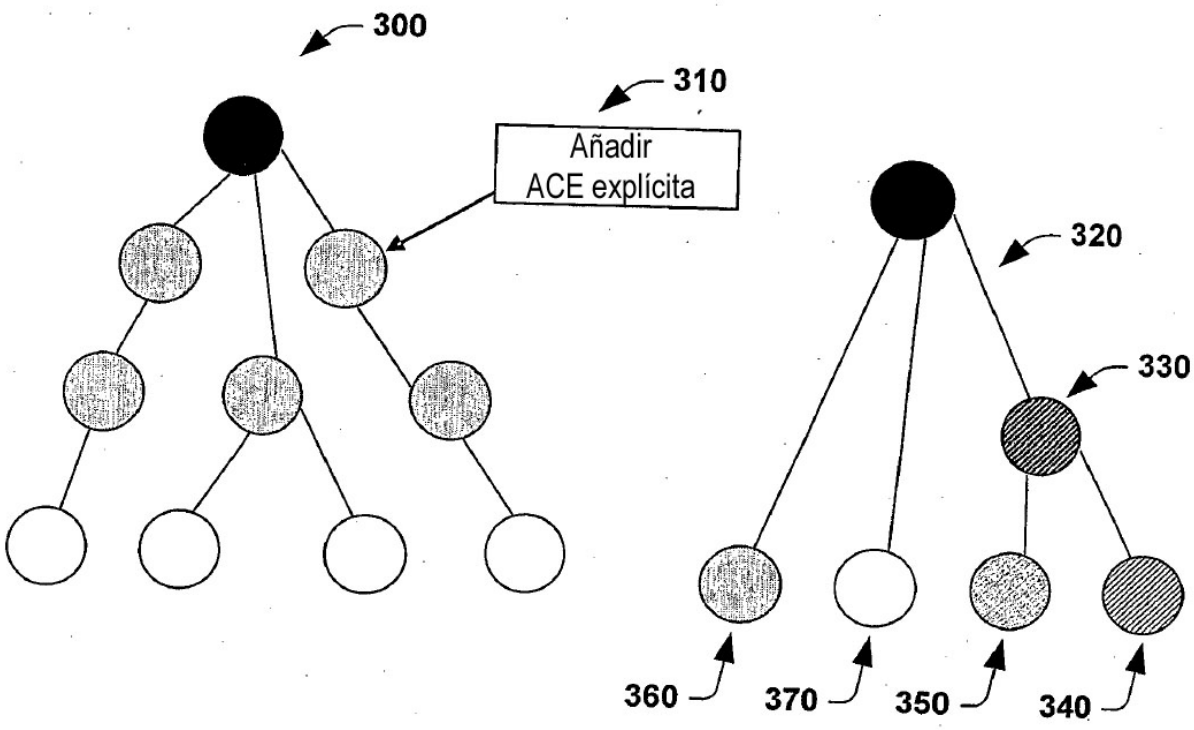

FIG. 3

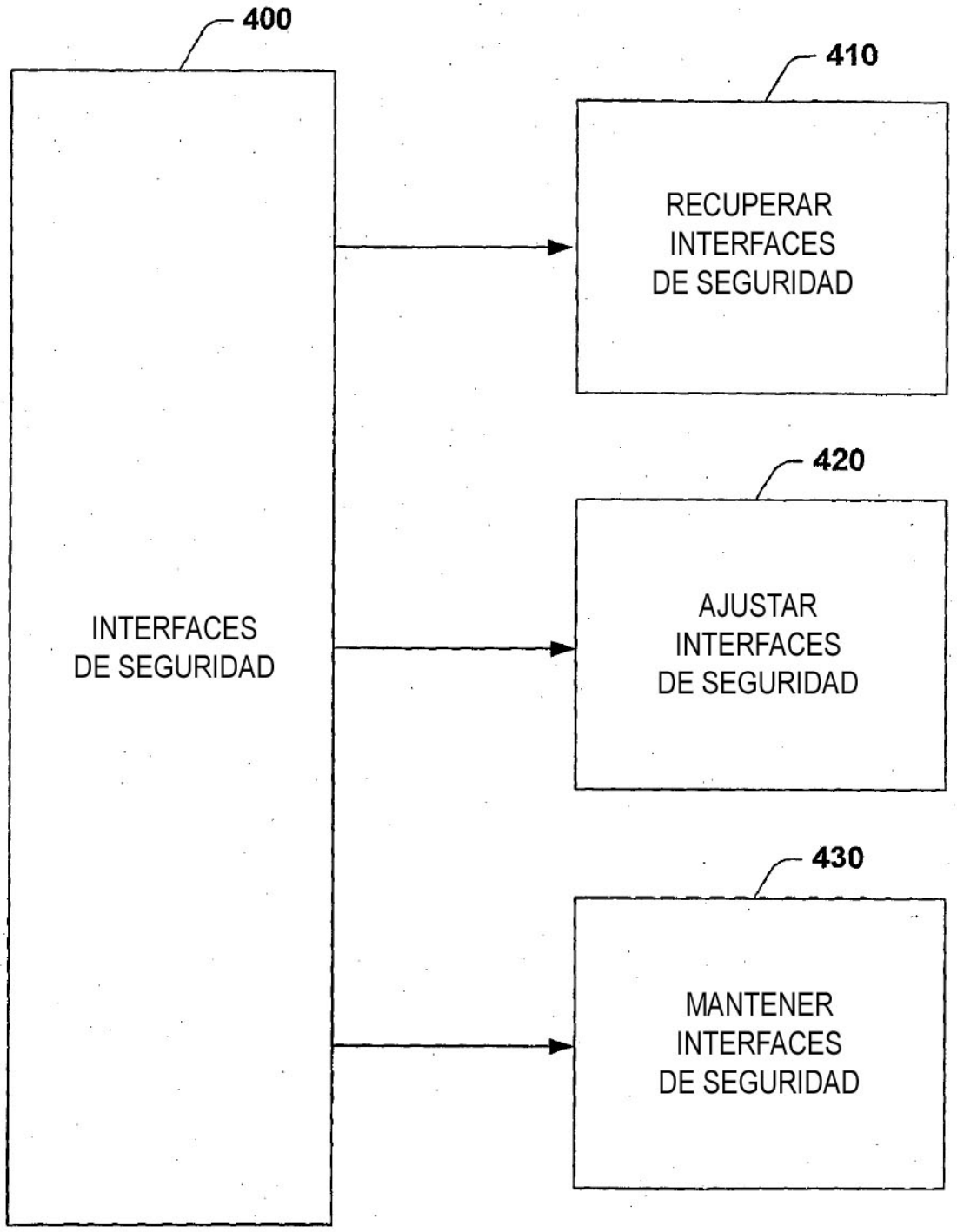
400
INTERFACES
DE SEGURIDAD
410
RECUPERAR
INTERFACES
DE SEGURIDAD
420
AJUSTAR
INTERFACES
DE SEGURIDAD
430
MANTENER
INTERFACES
DE SEGURIDAD

FIG. 4

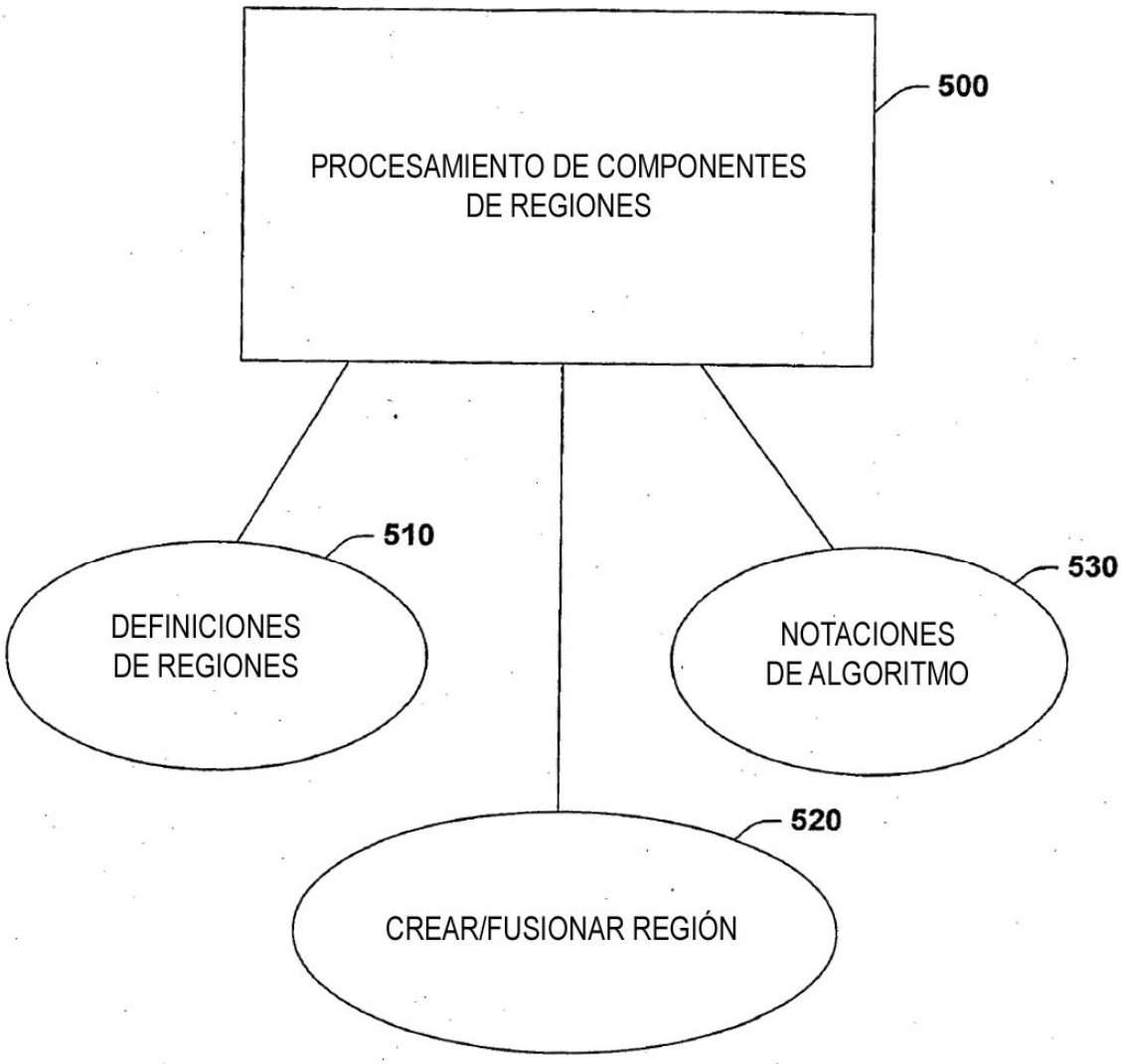
PROCESAMIENTO DE COMPONENTES DE REGIONES
500
DEFINICIONES DE REGIONES
510
NOTACIONES DE ALGORITMO
530
CREAR/FUSIONAR REGIÓN
520

FIG. 5

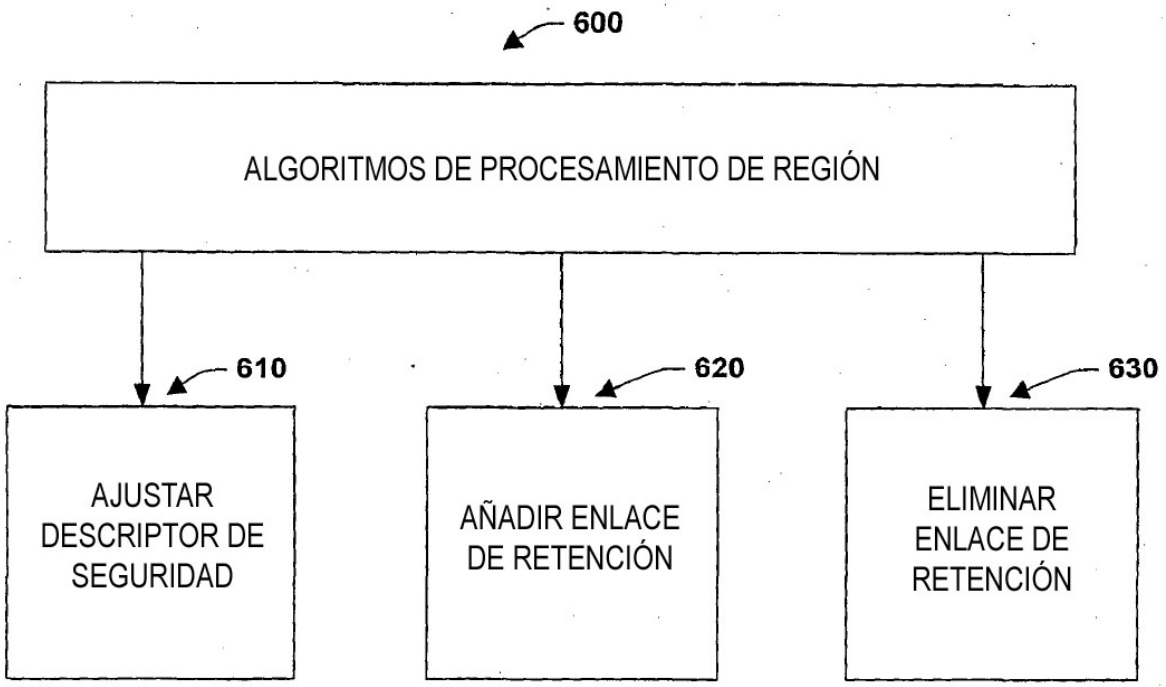
600
ALGORITMOS DE PROCESAMIENTO DE REGIÓN
610
620
630
AJUSTAR DESCRIPTOR DE SEGURIDAD
AÑADIR ENLACE DE RETENCIÓN
ELIMINAR ENLACE DE RETENCIÓN

FIG. 6

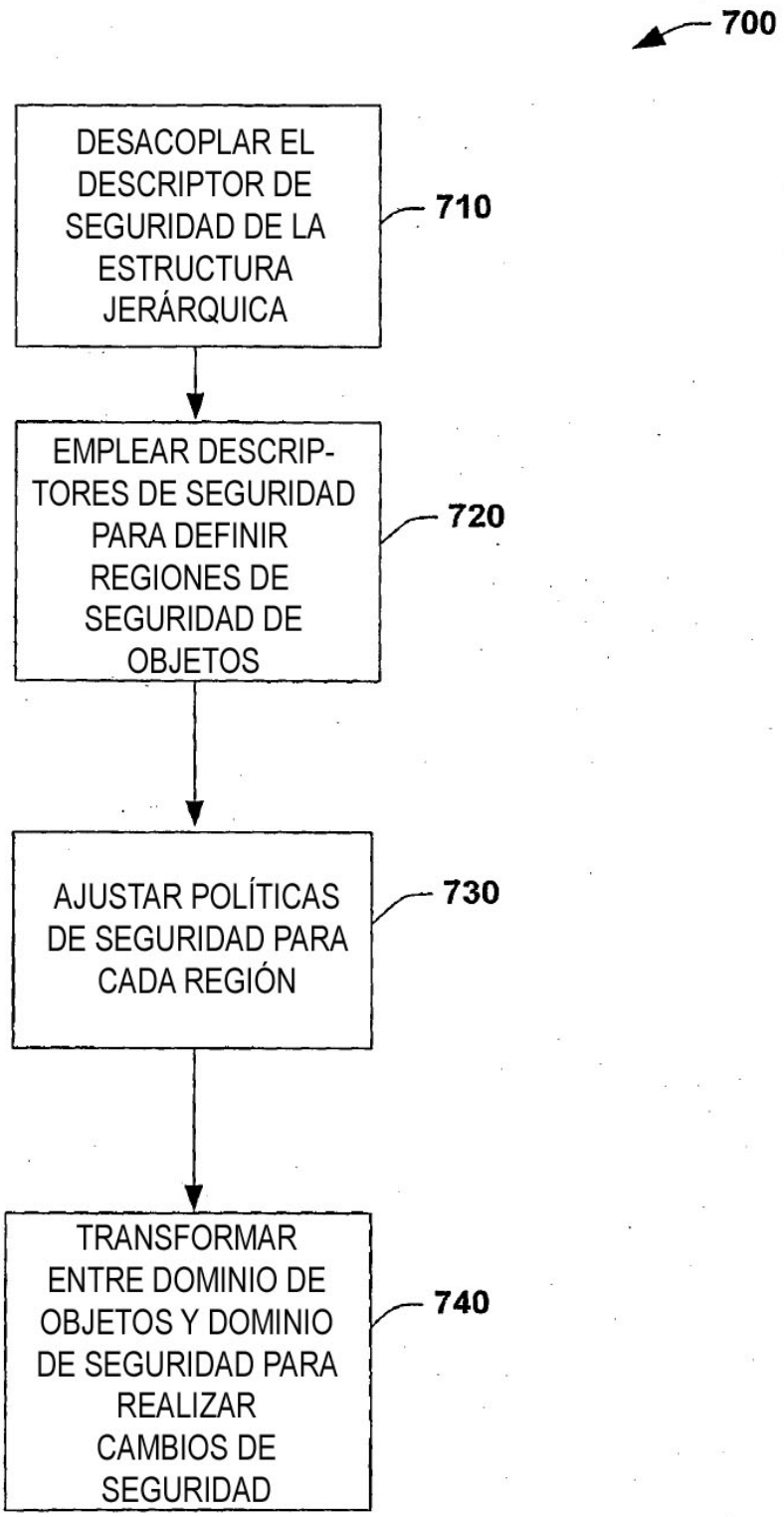
700
DESACOPLAR EL DESCRIPTOR DE SEGURIDAD DE LA ESTRUCTURA JERÁRQUICA
710
EMPLEAR DESCRIP-TORES DE SEGURIDAD PARA DEFINIR REGIONES DE SEGURIDAD DE OBJETOS
720
AJUSTAR POLÍTICAS DE SEGURIDAD PARA CADA REGIÓN
730
TRANSFORMAR ENTRE DOMINIO DE OBJETOS Y DOMINIO DE SEGURIDAD PARA REALIZAR CAMBIOS DE SEGURIDAD
740

FIG. 7

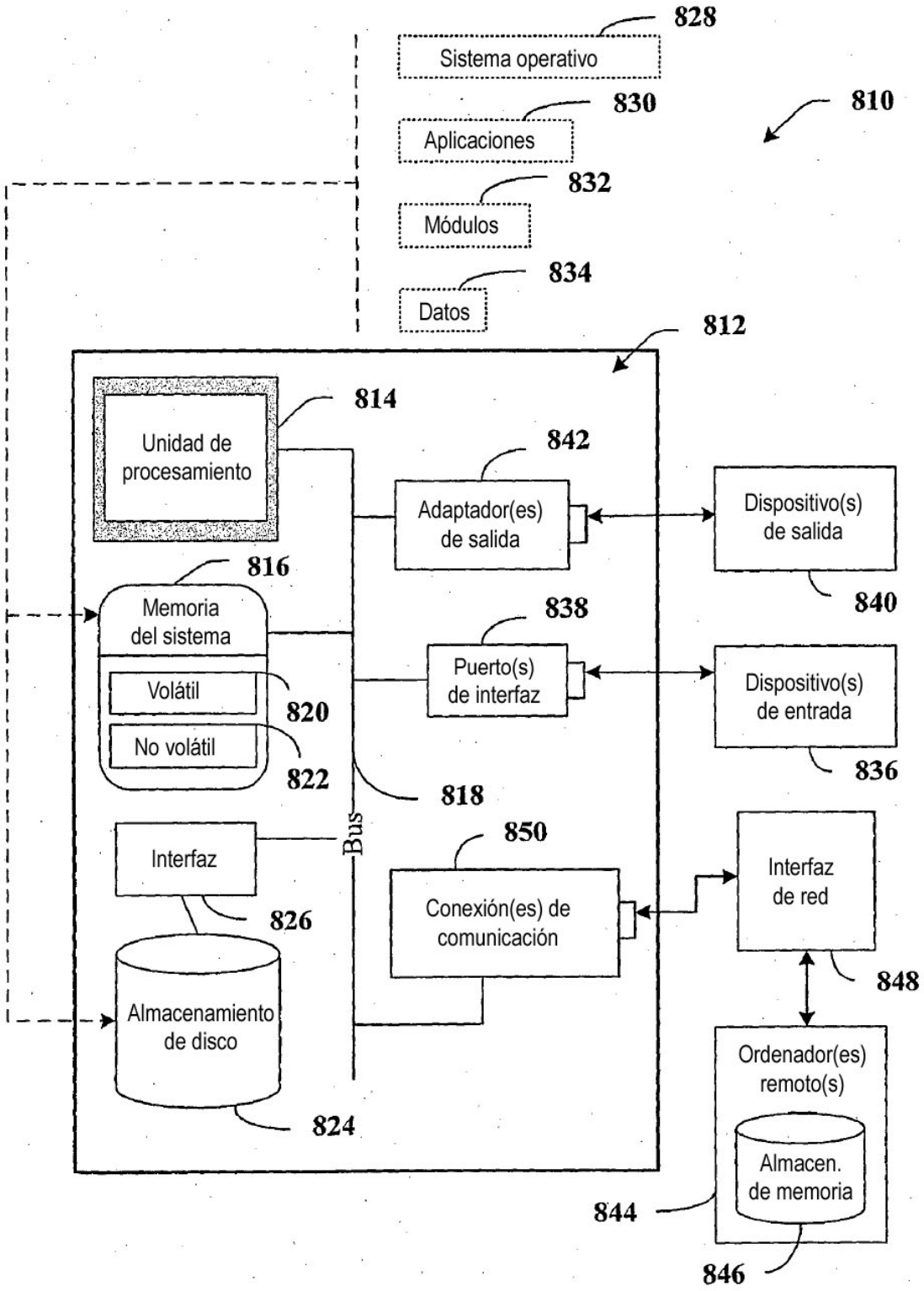
828
Sistema operativo
830
Aplicaciones
832
Módulos
834
Datos
810
812
814
Unidad de procesamiento
842
Adaptador(es) de salida
Dispositivo(s) de salida
840
816
Memoria del sistema
Volátil
820
No volátil
822
838
Puerto(s) de interfaz
Dispositivo(s) de entrada
836
818
Bus
Interfaz
826
850
Conexión(es) de comunicación
Interfaz de red
848
Almacenamiento de disco
824
Ordenador(es) remoto(s)
Almacen. de memoria
844
846

FIG. 8

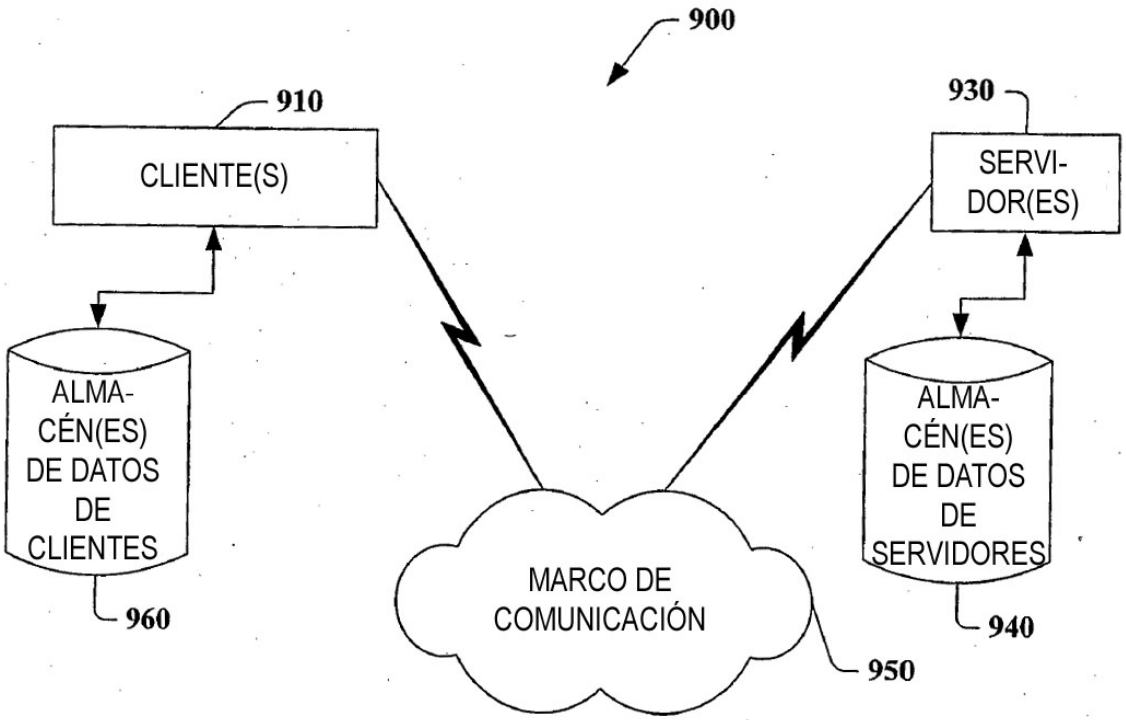
900
910
CLIENTE(S)
930
SERVI-
DOR(ES)
ALMA-
CÉN(ES)
DE DATOS
DE
CLIENTES
960
MARCO DE
COMUNICACIÓN
950
ALMA-
CÉN(ES)
DE DATOS
DE
SERVIDORES
940

FIG. 9