(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 627 855**

(51) Int. CI.:

G06F 21/14 (2013.01)
G06F 21/62 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **01.02.2006 E 06101144 (1)**

(97) Fecha y número de publicación de la concesión europea: **22.03.2017 EP 1696330**

(54) Título: **Mecanismos de capacidad de descubrimiento y enumeración en un sistema de almacenamiento jerárquicamente seguro**

(30) Prioridad:

28.02.2005 US 657536 P
28.06.2005 US 168589

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
31.07.2017

(73) Titular/es:

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

(72) Inventor/es:

HUNTER, JASON T.;
DUBHASHI, KEDARNATH A. y
SKARIA, SIMON

(74) Agente/Representante:

CARPINTERO LÓPEZ, Mario

DESCRIPCIÓN

Mecanismos de capacidad de descubrimiento y enumeración en un sistema de almacenamiento jerárquicamente seguro

Antecedentes

Los sistemas de almacenamiento de manera tradicional usan una jerarquía de contención para organizar unidades de almacenamiento. De acuerdo con estos sistemas, un contenedor y por lo tanto, intrínsecamente las unidades de datos mantenidas en el contenedor, son asegurables de manera independiente para facilitar el aprovisionamiento de acceso a los principales. Los sistemas convencionales ofrecen capacidad de descubrimiento a través del recorrido que podría limitar el acceso a datos después de encontrar un contenedor que no sea accesible para el principal.

Estos sistemas sufren de al menos las siguientes limitaciones. Una limitación es que un principal no puede visualizar el conjunto global de datos para el que tiene acceso. En otras palabras, después de representar un conjunto global de datos, si se encuentra un contenedor mediante el cual un usuario no tiene acceso, los contenidos (por ejemplo, unidades de datos) de este contenedor podrían no representarse. Considérese una situación donde existe una sub-carpeta o sub-contenedor en un contenedor con restricciones de acceso colocada en el principal. En este escenario, el principal podría no visualizar (por ejemplo, descubrir) o acceder a los contenidos de la sub-carpeta incluso si están presentes permisos adecuados. Esta capacidad de descubrimiento restrictiva es debido a la ausencia de permisos adecuados para acceder a la carpeta padre.

Otra limitación de los sistemas tradicionales es que un principal no puede operar en todos los datos a la vez. Por ejemplo, una restricción para una operación tal como “conceder acceso a FABRIKAM\alice para todos los datos en la estructura similar a árbol enraizada en un nodo dado” no sería posible ya que pueden estar presentes restricciones que limitarían el acceso a algunos de los datos en la estructura similar a árbol. En algunos sistemas tradicionales, tal operación se efectúa en el contexto del usuario en lugar de en un contexto de sistema.

Otra limitación más de algunos sistemas convencionales es que acceder a los datos requiere que estén presentes permisos adecuados para todos los contenedores desde el punto de conexión al padre inmediato de la unidad de datos además de permisos de acceso en la unidad de almacenamiento. En otras palabras, en algunos sistemas, incluso si la ruta de fichero directa de los datos es conocida, el permiso para acceder a los datos puede estar restringido si no existen permisos de acceso desde el punto de conexión al padre inmediato donde se almacenan los datos.

Otra limitación más es que, para la enumeración eficaz en el modelo de sistema de ficheros existente, los sistemas de almacenamiento tradicionales distinguen entre datos y metadatos. Para tipos de usuarios finales avanzados, esta separación crea dificultad para reconocer la distinción entre metadatos y datos.

El documento US 6 158 007 A desvela un procedimiento y un sistema que permiten aplicar control de acceso basándose en una identidad autenticada de usuarios cuando se accede a datos en una base de datos. Adicionalmente, cuando se almacenan mensajes con protecciones particulares, el objeto puede organizarse fácilmente en estructuras jerárquicas para formar árboles objeto, de modo que las características de un objeto se heredan por algunos o todos sus objetos descendientes. Mediante el uso de tales árboles de objeto, se hace innecesario asociar explícitamente una política de seguridad con cada uno y todos los objetos. Un objeto hijo puede simplemente heredar la política de seguridad, y por lo tanto la lista de control de acceso (ACL) y la calidad de protección (QOP) de un objeto padre.

El documento US 2003/188198 A1 proporciona procedimientos, aparatos y programas informáticos para aplicar controles de acceso a operaciones de control en recursos de sistemas de procesamiento de datos organizados jerárquicamente. Puede establecerse un número de diferentes alcances de aplicabilidad en asociación con un control de acceso, tal como una ACL, y esto determinará la capacidad de heredar, la no capacidad de heredar o la capacidad limitada de heredar el control de acceso para recursos en la jerarquía. Cuando se recibe una solicitud para realizar una operación, los controles de acceso para la rama relevante de la jerarquía se procesan para determinar un control de acceso aplicable que tiene en cuenta los atributos de capacidad de heredar que se han establecido para controles de acceso individuales.

Es el objeto de la presente invención proporcionar un sistema y procedimiento que facilita acceder a datos almacenados de acuerdo con una representación jerárquica.

Este objeto se resuelve mediante la materia objeto de las reivindicaciones independientes.

Las realizaciones se proporcionan en las reivindicaciones dependientes.

Sumario

Un aspecto de la presente invención es un aparato como se define en la reivindicación independiente 1. Otro aspecto de la invención es un procedimiento como se define en la reivindicación independiente 14. Otro aspecto

más de la invención se refiere a un medio de almacenamiento que almacena un programa que implementa el procedimiento como se define en la reivindicación 18. Se especifican realizaciones adicionales de la invención en las respectivas reivindicaciones dependientes adjuntas.

Breve descripción de los dibujos

La Figura 1 ilustra un diagrama de bloques de componente general de un sistema que facilita la capacidad de descubrimiento de datos en un sistema de almacenamiento seguro jerárquico de acuerdo con un aspecto de la invención.

La Figura 2 ilustra un diagrama de bloques de un sistema que incluye una tabla de instancia única y una tabla de descriptor de seguridad de acuerdo con un aspecto de la invención.

La Figura 3 ilustra un sistema que clasifica elementos en un sistema de tipo como instancias de tipos de contenedor genérico y tipos de elemento compuesto de acuerdo con un aspecto.

La Figura 4 ilustra un diagrama de bloques de un sistema que tiene un componente de almacenamiento y un componente de cliente en lados opuestos de un límite de confianza de acuerdo con un aspecto de la invención.

La Figura 5 ilustra una metodología de inicialización de acuerdo con un aspecto de la invención.

La Figura 6 es un diagrama relacional que ilustra que las operaciones que consultan las vistas pueden operar en el contexto de usuario donde puede aplicarse el control de acceso para sentencias de selección mediante seguridad de nivel de fila de acuerdo con un aspecto de la invención.

La Figura 7 es un diagrama de bloques de un sistema que emplea mecanismos basados en inteligencia artificial de acuerdo con un aspecto de la invención.

La Figura 8 ilustra un diagrama de bloques de un ordenador operable para ejecutar la arquitectura desvelada.

La Figura 9 ilustra un diagrama de bloques esquemático de un entorno informático ejemplar de acuerdo con la invención objeto.

Descripción detallada

La invención se describe ahora con referencia a los dibujos, en los que se usan números de referencia similares para hacer referencia a elementos similares a lo largo de todo el documento. En la siguiente descripción, para fines de explicación, se exponen numerosos detalles específicos para proporcionar un entendimiento minucioso de la invención objeto. Puede ser evidente, sin embargo, que la invención puede ponerse en práctica sin estos detalles específicos. En otros casos, estructuras y dispositivos bien conocidos se muestran en forma de diagrama de bloques para facilitar describir la invención.

Como se usa en esta solicitud, los términos “componente” y “sistema” se pretende que hagan referencia a una entidad relacionada con ordenadores, ya sea hardware, una combinación de hardware y software, software, o software en ejecución. Por ejemplo, un componente puede ser, pero sin limitación, un procedimiento que se ejecuta en un procesador, un procesador, un objeto, un ejecutable, un subprocedimiento de ejecución, un programa y/o un ordenador. Por medio de ilustración, tanto una aplicación que se ejecuta en un servidor como el servidor pueden ser un componente. Uno o más componentes pueden residir en un procedimiento y/o hilo de ejecución, y un componente puede localizarse en un ordenador y/o distribuirse entre dos o más ordenadores.

Como se usa en el presente documento, el término “inferir” o “inferencia” hacen referencia en general al procedimiento de razonamiento acerca de o estados de inferencia del sistema, entorno y/o usuario a partir de un conjunto de observaciones según se capturan mediante eventos y/o datos. La inferencia puede emplearse para identificar un contexto o acción específicos, o puede generar, por ejemplo, una distribución de probabilidad sobre estados. La inferencia puede ser probabilística, es decir, el cálculo de una distribución de probabilidad sobre estados de interés basándose en una consideración de datos y eventos. La inferencia también puede hacer referencia a técnicas empleadas para componer eventos de nivel superior a partir de un conjunto de eventos y/o datos. Tales resultados de inferencia en la construcción de nuevos eventos o acciones a partir de un conjunto de eventos observados y/o datos de eventos almacenados, estén o no los eventos correlacionados en proximidad temporal cercana, y provengan los eventos y datos de una o varias fuentes de eventos y datos.

Los aspectos de esta invención están relacionados con sistemas informáticos y más particularmente la capacidad de descubrimiento de datos mantenidos en un sistema o sistemas de almacenamiento jerárquicamente seguro. Como se ha descrito anteriormente, los sistemas de almacenamiento tradicionales tienen limitaciones con respecto a los mecanismos de capacidad de descubrimiento relacionados con seguridad. Para este fin, los sistemas de ficheros orientados a bases de datos emergentes pueden soportar consultas avanzadas y proporcionar tipos de usuario final esquematizados para unidades de datos comunes (por ejemplo, contactos). Estos tipos de usuario final esquematizados facilitan y pueden mejorar la interoperabilidad de las aplicaciones con respecto a datos.

La invención objeto tiene en cuenta una representación jerárquica de datos. Más particularmente, esta invención tiene en cuenta que los datos pueden “almacenarse en compartimentos” en diferentes carpetas y posteriormente colocarse en diferentes contenedores. Los usuarios pueden emplear estos contenedores para organizar sus datos. Por ejemplo, los datos pueden organizarse (por ejemplo, almacenarse en compartimentos) en categorías tales como imágenes, música, documentos, etc. Adicionalmente, estas categorías pueden organizarse adicionalmente en contenedores estableciendo de esta manera una representación jerárquica de los datos. A modo de ejemplo, dentro

de las imágenes, podría haber imágenes de “mi familia”, “mis vacaciones”, “mi boda”, etc. Así como pueden existir subcategorías de acuerdo con la jerarquía.

De acuerdo con esta representación jerárquica, la invención puede facilitar asociar una política de seguridad (por ejemplo, descriptor de seguridad) con cada objeto. Se apreciará que un objeto puede ser cualquier elemento de datos contenido en un contenedor así como el propio el contenedor. También, cada objeto puede representarse en una fila individual de una tabla. Esta representación basada en filas puede entenderse mejor tras un análisis de las figuras que siguen.

En un aspecto, el descriptor de seguridad puede posibilitar el aprovisionamiento de estos objetos para acceso de datos. A modo de ejemplo, de acuerdo con un aspecto de la invención, una política de seguridad puede facilitar establecer una carpeta “mis vacaciones” para permitir el acceso por cualquiera en un grupo, “mi familia.” Así como, en “mis vacaciones” un usuario puede limitar adicionalmente el acceso a ciertos miembros de “mi familia” para acceder a una subcarpeta (por ejemplo, “mi viaje a Seattle”).

De acuerdo con sistemas convencionales, la exploración accesible de un almacenamiento de datos finaliza en cualquier punto cuando se alcanza una carpeta para la que el usuario no tiene acceso de enumeración. Considerando una jerarquía donde F1 contiene F2 que contiene F3 - el momento en el que el usuario alcanza F2 donde no se concede permiso, el usuario no tendrá la capacidad de ver los datos en F3. Incluso aunque el usuario pueda acceder a F3, los sistemas convencionales prohibirán la capacidad de descubrimiento puesto que F3 está contenido en F2 para el que no están presentes permisos - esto es una limitación. La invención objeto posibilita que un usuario tenga acceso uniforme para explorar (por ejemplo, descubrir) y/o representar de esta manera permitiendo el empleo de todos los datos en un almacenamiento de datos mediante el cual se conceden y están presentes los permisos. Como se ha descrito anteriormente, este acceso uniforme puede facilitarse mediante una política de seguridad asociada con cada objeto en un almacenamiento de datos. Como se entenderá, cada política de seguridad puede asociarse con un elemento de nivel de fila.

Los sistemas de ficheros tradicionales emplean dos modos de acceso para recuperar ficheros. En primer lugar, estos sistemas facilitan un procedimiento de descubrimiento limitado mediante el cual un usuario puede descubrir elementos de datos para los que existen permisos de seguridad adecuados. El otro es un mecanismo de acceso directo mediante el cual un usuario puede acceder a un fichero si se conoce la ruta completa y está presente el permiso para acceso.

Además de los dos modos distintos, la invención objeto puede emplear un tercer modo que es un modo de consulta (por ejemplo, filtración de almacenamiento de datos) que permite acceso y descubrimiento basándose en credenciales de seguridad. A diferencia de los sistemas tradicionales, la invención objeto puede proporcionar un mecanismo para consultar todos los datos basándose en una propiedad especificada definida así como para operar en esos datos. Con esta invención, siempre que estén presentes credenciales de acceso, los datos pueden descubrirse y operarse según se desee.

De acuerdo con lo mismo, la invención objeto puede posibilitar una política de seguridad (por ejemplo, descriptor de seguridad) que puede establecerse en la raíz de una estructura similar a árbol (por ejemplo, organización de datos jerárquica) y propagarse a través de la estructura similar a árbol de todos los hijos en la estructura. Se ha de entender que el descriptor de seguridad propagado puede basarse en la política de seguridad padre, política de seguridad hijo y/o el tipo del objeto. Puede emplearse lógica que efectúa la generación y propagación de una política de seguridad a través de una estructura similar a árbol. Como se describirá más adelante, pueden emplearse reglas basadas en lógica y/o inteligencia artificial para propagar una política de seguridad.

Considerando un escenario donde un usuario crea un nuevo elemento. En este escenario, podría haber ciertas políticas de seguridad (por ejemplo, descriptores) del padre que pueden heredarse o combinarse en el hijo. En un aspecto, un usuario puede tener una carpeta (por ejemplo, contenedor) con permisos y cuando se crea un objeto, los permisos para el objeto puede suponerse que son los mismos. Como alternativa, los permisos propagados al objeto recién creado pueden determinarse de manera inteligente basándose tanto en los permisos para la carpeta así como los permisos para el objeto. Los anteriores son ejemplos de heredar de acuerdo con aspectos de la innovación novedosa.

Se apreciará que, en sistemas de ficheros tradicionales, esta propagación no es posible. En su lugar, para cambiar permisos de acuerdo con los sistemas convencionales, un administrador debe pasar a través de cada hijo de una estructura similar a árbol y cambiar los permisos según sea aplicable. Al contrario, de acuerdo con aspectos de esta invención, cuando se cambia (o establece) un permiso de raíz, el permiso puede propagarse automáticamente a toda la estructura similar a árbol, incluyendo los hijos.

Es importante observar que, en algunos sistemas tradicionales, los permisos de seguridad podrían propagarse únicamente en el “contexto del usuario” en el momento de la actualización. Aunque hay situaciones donde los permisos pueden cambiar en un momento más tarde, los sistemas convencionales no pueden actualizar automáticamente estos permisos.

La invención objeto puede propagar permisos en el “contexto del sistema”. Por lo tanto, incluso si un usuario no tiene permiso a una carpeta intermedia, si están presentes permisos para una sub, sub-sub, etc., estructura similar a árbol, estos permisos pueden propagarse de acuerdo con la invención. Este aspecto se entenderá mejor considerando el ejemplo F1, F2 y F3 anteriormente mencionado.

Continuando con el ejemplo, incluso si no están presentes permisos para F2, si existen permisos para F3, los permisos pueden propagarse de F1 a F3. A diferencia de sistemas de ficheros anteriores que distinguen entre atributos (por ejemplo, nombre del fichero, tamaño, fecha creada) y datos (por ejemplo, contenido del fichero), en sistemas de datos avanzados es difícil determinar entre un atributo y datos. Como tal, los “elementos” se crearon y usaron para conceder acceso a permisos en una base por “elemento” independientemente de que elemento de datos sea un atributo o datos. Por consiguiente, con respecto a la invención objeto, la gestión del modelo de seguridad puede simplificarse particularmente puesto que el sistema no tiene que rastrear dos permisos de seguridad separados. En su lugar, en un aspecto, únicamente se emplea un permiso de “lectura” o únicamente uno de “escritura” por elemento en lugar de emplear dos permisos de “lectura” y dos permisos de “escritura” por elemento.

Como resultado, la invención puede facilitar a un usuario ver una abstracción de todos los datos para los que están presentes permisos. Estas vistas pueden definirse a través de todo el almacenamiento y representarse posteriormente a un usuario. La vista puede definirse como una intersección de los elementos visibles desde un punto de conexión y el conjunto de permisos de seguridad permitidos. Como resultado, un usuario puede ver y/o acceder a elementos por debajo de un punto de conexión para el que el usuario tiene permisos de seguridad para visión y/o acceso.

Haciendo referencia inicialmente a la Figura 1, se muestra un sistema 100 que facilita representar una representación de contenido de almacenamiento de fichero. En general, el sistema 100 puede incluir un componente 102 de consulta y un componente 104 de seguridad de nivel de fila. En la operación, el componente 102 de consulta, junto con el componente 104 de seguridad de nivel de fila pueden identificar elementos en un componente 106 de datos que satisfacen una política de seguridad o permiso. Una vez identificado, el conjunto de datos resultante puede representarse a un usuario y/o aplicación. Por ejemplo, como se ha descrito anteriormente, la invención puede representar el conjunto resultante mediante una pantalla a un usuario.

Con referencia ahora a la Figura 2, se muestra un diagrama de bloques más detallado del componente 104 de seguridad de nivel de fila. En particular, el componente 104 de seguridad de nivel de fila puede incluir una tabla 202 de descriptor de seguridad y una tabla 204 de instancia única. Cada una de estas tablas de describirá en mayor detalle más adelante.

El componente 104 de seguridad puede proporcionar una compresión de la seguridad de nivel de fila. Cuando el usuario se conecta a una compartición (por ejemplo, el componente 106 de datos), pueden definirse definiciones de vista implícitas para cada uno de los tipos de datos en el ámbito de la conexión. Para añadir contexto a la invención, a continuación hay una definición de vista ejemplar para un tipo “Contacto”.

```
CREATE VIEW [System.Storage.Contacts.Store].[Contact] AS
   SELECT ItemId, TypeId, NamespaceName, ContainerId,
ItemSyncMetadata,
      TREAT(Item AS [System.Storage.Contacts.Store].[Contact]) AS
Item, PathHandle,
      EntityState, ObjectSize, ChangeInformation, PromotionStatus
   FROM [System.Storage.Store].[Table!Item]
WHERE Item IS OF ([System.Storage.Contacts.Store].[Contact])
   AND (@@ITEM_DOMAIN_IS_ROOT = 1
   OR (PathHandle >= @@ITEM_DOMAIN AND PathHandle <
@@ITEM_DOMAIN_LIMIT))
```

Cada elemento se almacena como una fila en las tablas (202, 204) de entidad. La expresión ejemplar anterior puede efectuar la filtración de los tipos Contacto a partir del ámbito global de los elementos en el almacenamiento. Implícito a esta filtración es la dimensión de control de acceso donde un usuario vería únicamente aquellos elementos que son legibles de acuerdo con los descriptores de seguridad en la fila correspondiente.

En este ejemplo, una definición de vista puede incluir la cláusula “WHERE” anteriormente identificada que restringe una vista a elementos que son contactos. El resto del ejemplo puede restringir acceso a elementos desde el punto de conexión. Se ha de entender que la definición de vista anterior no incluye la definición de seguridad.

Como se ha descrito anteriormente, el mecanismo de seguridad es una función de la seguridad de nivel de fila almacenada en las tablas (202, 204). Este mecanismo se aplica en el nivel de tabla subyacente de la vista y tiene efectos de propagación sobre la vista. Cuando se activa la seguridad en una base por fila, las filas para las que un usuario no tiene acceso de lectura no aparecen en el conjunto resultante proporcionado por el componente 102 de consulta.

En un modelo de sistema de ficheros, cada “elemento” está en una fila, y cada fila tiene seguridad asociada con ella. El mecanismo 104 de seguridad de nivel de fila restringe que las filas aparezcan en los resultados para aquellas filas que un usuario no tiene acceso de lectura. La vista, dada una definición transmitida componente 102 de consulta, (como en el ejemplo anterior) puede restringir la representación (por ejemplo, visualización) basándose al menos en parte en el punto de conexión. Por lo tanto, el conjunto resultante, puede ser la intersección de estas dos restricciones. Se apreciará que estos mecanismos de seguridad pueden tener lugar implícitos a la definición de consulta. Como resultado, el usuario puede protegerse de cualquiera de las operaciones.

La invención objeto emplea un único mecanismo de generación de instancias que comprueba el descriptor de seguridad de cada fila en la tabla (por ejemplo, 204). Este único mecanismo de generación de instancias hace posible parecer que el sistema está realizando una comprobación a través de cada fila. Una única generación de instancias de descriptores de seguridad a través de las filas puede hacer eficaz la comprobación de este mecanismo. Se apreciará que pueden emplearse políticas de seguridad (por ejemplo, listas de control de acceso) en lugar de los descriptores de seguridad ejemplares. Por lo tanto, se ha de entender que estos aspectos novedosos adicionales se pretende que caigan dentro del alcance de esta invención y las reivindicaciones adjuntas a la misma. Adicionalmente, aunque se han mencionado anteriormente las ACL, se ha de entender que existen otros aspectos que emplean políticas de seguridad distintas. Estas políticas de seguridad distintas se pretende que caigan dentro del alcance de esta invención según se reivindica.

En la operación, se mantienen dos tablas (202, 204) - una tabla de descriptores 202 de seguridad y una tabla de instancia única del mapeo entre el hash (por ejemplo, SHA-1) del descriptor de seguridad y una identificación de descriptor de seguridad (SDID). Se apreciará que este SDID es un valor único. De acuerdo con la invención, la generación de instancias única se refiere a un mecanismo cuando, para cada descriptor de seguridad único en el almacenamiento, el sistema mantiene un mapa entre el SDID y un hash del descriptor de seguridad.

Por lo tanto, para cada fila, en lugar de almacenar un descriptor de seguridad, se almacena el SDID al que corresponde. En un aspecto, cuando un usuario crea un elemento, el usuario tiene una elección de proporcionar un descriptor de seguridad o dejarlo vacío. Si se deja vacío, el descriptor de seguridad puede heredarse del padre desde el que se crea el elemento. Cuando el usuario opta por proporcionar explícitamente un descriptor de seguridad, el sistema puede unir el descriptor explícitamente definido con el descriptor de seguridad del padre para crear uno.

Una vez que se realiza una determinación de cuál será el descriptor de seguridad en el nuevo elemento, se realizará una determinación de si ya existe. Si existe, se usará el existente. Si no existe, se grabará el nuevo.

Para determinar si existe un descriptor de seguridad, la invención hace referencia a la tabla 204 de instancia única que incluye un mapeo del descriptor de seguridad a un hash (por ejemplo, hash SHA-1) del descriptor de seguridad. Por lo tanto, para determinar si existe otro elemento con el mismo descriptor de seguridad, se calcula un hash del descriptor de seguridad objeto. El sistema a continuación consulta la tabla 204 de generación de instancias únicas para una fila, para ver si alguna fila contiene el mismo hash (por ejemplo, SHA-1) del descriptor de seguridad. Si se encuentra una coincidencia, hay una alta probabilidad de que exista.

A continuación, se realiza una comparación del descriptor de seguridad real para verificar si existe el descriptor de seguridad. Si el descriptor de seguridad real no es el mismo, el sistema almacena el descriptor de seguridad de manera independiente. Se ha de apreciar que el sistema únicamente se basa en el algoritmo hash (por ejemplo, SHA-1) para garantizar la no singularidad. En otras palabras, si el valor hash no coincide con un valor hash en la tabla 204 de instancia única, puede realizarse una determinación de que no existe el descriptor de seguridad.

Hay tres propiedades para un descriptor de seguridad - el hash (valor calculado matemáticamente basándose en el binario del descriptor de seguridad), el propio descriptor de seguridad (binario), y el SDID (valor entero que apunta al descriptor de seguridad). Para cada fila, el sistema almacena el ID de esa fila particular para el que es relevante el descriptor de seguridad. A continuación, en la tabla 204 de instancia única, el sistema realiza el mapeo entre el hash (por ejemplo, SHA-1) y el SDID. En la tabla 202 de descriptor de seguridad, el sistema realiza el mapeo entre el SDID y el binario.

Por lo tanto, la tabla 204 de instancia única y la tabla 202 de descriptor de seguridad proporcionan juntas un mapeo completo desde un hash SHA-1 a SDID al binario. De manera eficaz, estas dos tablas (202, 204) pueden usarse para realizar una única comprobación de generación de instancias.

Un descriptor de seguridad puede tener la siguiente forma lógica:

```
O:owner_sid
G:group_sid
D:dacl_flags(ace1) (ace2)... (acen)
S:sacl_flags(ace1) (ace2)... (acen)
```

En el ejemplo anterior, O: identifica el propietario, G: identifica el grupo, D: identifica la Lista de Control de Acceso Discrecional (DACL) (la sección del descriptor de seguridad en el alcance de la divulgación) y S: identifica la Lista de

Control de Acceso de Sistema (SACL). DACL es una colección de Entradas de Control de Acceso (ACE) - cada una puede tomar la siguiente forma. ace_type;ace_flags;rights; account_sid

Puede concederse o denegarse acceso a un principal dado a elementos específicos. Por consiguiente, los elementos denegados pueden filtrarse implícitamente de las vistas del usuario. Un motor de filtración o componente 102 de consulta puede explorar todos los elementos en el almacenamiento agnóstico a cualquier semántica de contenedor y producir un conjunto uniforme evitando de esta manera las limitaciones de los recorridos en los sistemas de ficheros tradicionales.

Las dos tablas (202, 204) internas pueden usarse para facilitar el almacenamiento y control de acceso en el sistema. En un aspecto ejemplar, el sistema puede emplear una tabla 204 [System.Storage.Store].[Table!SecurityDescriptorSingleInstance] (por ejemplo, tabla de instancia) y una tabla 202 Sys.security_descriptors (por ejemplo, tabla de descriptor de seguridad). La tabla 202 Sys.security_descriptors es una vista de catálogo de descriptores de seguridad. Estos descriptores pueden crearse o borrarse usando primitivas de lenguaje de definición de datos (DDL) proporcionadas mediante SQL Server. La tabla 204 de instancia única puede ser clave para una unidad de procesamiento central (CPU) y optimizaciones de memoria en el sistema.

De acuerdo con un aspecto, puede ser común que un número de elementos significativo comparta la misma política de seguridad o descriptor. En un ejemplo, el tamaño máximo de una lista de control de acceso (ACL) es 64 KB por lo tanto un descriptor de seguridad dado puede estar en el orden de 128 KB. Se apreciará que puede ser ineficaz almacenar un valor de este tamaño con cada elemento dado su potencialmente alto grado de coincidencia. Por lo tanto, cada descriptor de seguridad único puede almacenarse en la tabla 202 Sys.security_descriptors y puede mantenerse un mapeo entre el descriptor y su hash SHA-1 en la tabla 204 de instancia única. Como se ha establecido anteriormente, un SHA-1 no garantiza singularidad de las salidas, pero una colisión es extremadamente improbable dado su gran intervalo de salida (por ejemplo, 2^160). Puesto que la tabla 204 de instancias puede tener una naturaleza auto-regenerativa, puede garantizar que el sistema pueda auto-recuperarse de corrupción o inconsistencias.

Las tablas Elemento/Extensión/Fragmento/Enlace tienen una entrada para el SDID que puede marcarse con el atributo SECURITY. Esto puede asegurar que todos los accesos de lectura a estas tablas y cualquier visión creada en la parte superior de estas tablas se sometan a una comprobación de acceso que solicite (FILE_READ_DATA | FILE_READ_ATTRIBUTES). Las filas en las tablas Extensión de elemento, Enlace y Fragmento de elemento tienen el mismo descriptor de seguridad que la fila correspondiente en la tabla Elemento.

El mecanismo anteriormente descrito puede considerarse que está en el núcleo de un modelo de autorización en la ruta de lectura para sistemas de ficheros emergentes. Cualquier modelo de autorización puede intrínsecamente basarse en un modelo de autenticación. En un ejemplo, cuando un usuario se conecta al almacenamiento, el usuario puede autenticarse (por ejemplo, considerarse confiable) usando los mecanismos de autenticación del sistema operativo preferido (por ejemplo, NTLM (Gestor de LAN de NT), Kerberos). El resultado neto de la autenticación puede ser un testigo de seguridad que representa al usuario que está accediendo al sistema de ficheros. Este testigo puede usarse posteriormente para realizar decisiones de autorización para el principal.

De acuerdo con otro aspecto de la invención, los elementos asegurados usando seguridad de nivel de registro (RLS) o de fila también pueden protegerse de la cuenta de servicio de almacenamiento. Para evaluación de seguridad, la cuenta de servicio puede considerarse como cualquier otra cuenta del estilo NT. Aunque esto puede garantizar particularmente semánticas de seguridad uniformes, proporciona problemas interesantes en la ruta de actualización. Por ejemplo, considérese que un usuario intenta crear un elemento con un nombre de espacio de nombres dado. Los nombres de espacio de nombres en sistemas de ficheros emergentes están garantizados para que sean únicos en su carpeta que lo contiene, proporcionado un sistema de nomenclatura desambiguo. Durante las operaciones de creación, el sistema garantiza esta singularidad asegurando la no existencia de otros elementos en la misma carpeta con el mismo nombre de espacio de nombres.

En este escenario, un elemento ya puede existir en la carpeta con permisos de acceso denegados a la cuenta de servicio. Esta invención puede tratar este problema usando un mecanismo de firma. Las primitivas de actualización que requieren acceso global al almacenamiento pueden firmarse con certificados que conceden privilegio “RLS exento”. A partir del contexto de una primitiva de este tipo, el sistema puede consultar el almacenamiento y se evitará la seguridad de nivel de fila en este caso.

Como se ha descrito anteriormente, los sistemas de ficheros tradicionales tienen una distinción entre atributos y datos para posibilitar las semánticas de recorrido. La ausencia de capacidad de descubrimiento y semánticas basadas en consulta indujeron un modelo donde se distinguen los atributos y datos para decisiones de control de acceso. La invención objeto proporciona acceso sin interrupciones a datos y atributos facilitando semánticas de todo o nada en el sistema de tipo.

A continuación se encuentra un análisis detallado de un modelo de seguridad de sistema de ficheros ejemplar. El análisis que sigue describe funcionalidad de componente en un número de escenarios distintos. Se ha de apreciar que estos escenarios descritos se proporcionan meramente para proporcionar contexto a la invención y no se pretenden para limitar la invención según se reivindica.

Haciendo referencia en primer lugar al modelo de seguridad de sistema de ficheros, en un aspecto, pueden organizarse los datos en un almacenamiento como un “elemento” que puede hacer referencia a la unidad más pequeña de consistencia en el sistema de ficheros. Un “elemento” puede asegurase, serializarse, sincronizarse, copiarse, respaldarse/restaurarse, etc., de manera independiente. Se apreciará que un elemento de sistema de ficheros puede describirse como una instancia de un tipo cuyo antecesor es el tipo System.Storage.Item, que es un tipo de entidad. Todos los elementos en el sistema de ficheros pueden almacenarse en una única extensión de elementos global. También, cada elemento puede tener un identificador único que se garantiza que sea único para todos los elementos en un almacenamiento de sistema de ficheros dado.

Haciendo referencia ahora a la Figura 3, se muestra un sistema 300. El sistema 300 es de acuerdo con el contexto de este análisis de seguridad mientras que los elementos en un sistema 302 de tipo pueden clasificarse como instancias de tipos 304 de contenedor genérico y tipos 306 de elemento compuesto. Pueden usarse contenedores 304 genéricos para modelar carpetas y cualquier otro compartimento de colección de datos jerárquica. Pueden usarse tipos 306 de elemento compuesto para modelar una única unidad lógica de datos para una aplicación. La instancias de este tipo proporcionan semánticas de todo o nada para operaciones de datos típicas como copiar, mover, sincronizar, etc. Ejemplos de esto último incluyen, pero sin limitación, mensajes de correo electrónico, imágenes, contactos, etc. Las instancias (indicadas mediante líneas discontinuas) de tipos 306 de elemento compuesto pueden clasificarse adicionalmente como elementos 308 respaldados de fichero (FBI) y elementos 310 respaldados no de fichero (nFBI). Se apreciará que un acceso del estilo Win32 está limitado semánticamente a FBI y a contenedores genéricos.

La siguiente jerarquía de contención (por ejemplo, estructura similar a árbol) se aplica a los elementos. Los contenedores 304 genéricos y elementos 306 compuestos pueden contener cualquier otro tipo de elemento que incluya contenedores genéricos. Los elementos en estos contenedores genéricos adicionales pueden asegurarse también de manera independiente. Los FBI 308 no pueden contener otros elementos y por lo tanto forman nodos hoja en la jerarquía.

Haciendo referencia ahora a la Figura 4, se apreciará que un sistema 400 de ficheros puede incluir dos componentes principales en lados opuestos de un límite 402 de confianza - un componente 404 de almacenamiento y un componente 406 de cliente. Como se ilustra, el componente 404 de almacenamiento puede incluir componentes de objeto 1 a N, donde N es un número entero. Los componentes de objeto 1 a N pueden hacerse referencia individual o colectivamente como componentes 408 de objeto. El componente 404 de almacenamiento que trata con el almacenamiento y recuperación del objeto 408 puede formar un subsistema de sistema de ficheros confiable entre el componente 404 de almacenamiento y el componente 406 de cliente.

El componente 406 de cliente que puede proporcionar semánticas de programación a la plataforma normalmente se ejecuta en los procedimientos del usuario. Se entenderá que los usuarios pueden autenticarse en el momento de conexión. Los objetos 408 recuperados (por ejemplo, elementos) pueden materializarse en el espacio de cliente. En un aspecto, no se aplican comprobaciones de seguridad o restricciones de acceso por el cliente en estos objetos 408. De acuerdo con la invención, el componente 404 de almacenamiento puede aplicar control de acceso (mediante componente 410 de control de acceso) cuando el contexto de programación hace persistir al componente 404 de almacenamiento. A continuación hay un análisis de la autenticación de usuario.

El sistema 400 de ficheros puede exponer la noción de un principal de seguridad que puede realizar las acciones frente a los elementos 408 contenidos en un almacenamiento 404 de sistema de ficheros. En aspectos de la invención, un principal de seguridad podría ser un usuario o un grupo de seguridad. Por consiguiente, el principal de seguridad podría representarse mediante un identificador de seguridad (SID).

Como se ilustra en la Figura 4, una conexión al servicio de sistema de ficheros es en el contexto de un principal de seguridad que está autenticado satisfactoriamente por el componente 410 de control de acceso. Se entenderá que la autenticación del sistema de ficheros (por ejemplo, mediante componente 410 de control de acceso) puede ser una derivada del mecanismo de autenticación del sistema operativo. Por ejemplo, una autenticación de sistema de ficheros puede ser una derivada de una autenticación del estilo Windows disponible en el modelo de seguridad de SQL (lenguaje de consulta estructurada). Por ejemplo, se apreciará que SQL ofrece otro mecanismo de autenticación integrado denominado autenticación de SQL que puede no soportarse en el sistema 400 de ficheros.

Continuando con el ejemplo, una conexión intentada por un usuario de estilo Windows puede autenticarse mediante el sistema 400 de ficheros mientras que aprovecha servicios de autenticación proporcionados por el estilo Windows tales como Kerberos, NTLM, etc. En el ejemplo, un usuario autenticado se mapea a una función “pública” en SQL que se usa para decisiones de autorización en el almacenamiento 404. En un aspecto, un administrador integrado (BA) se mapeará a administradores de SQL que conceden privilegios administrativos de SQL al BA. En un aspecto alternativo, la administración del sistema de ficheros puede integrarse únicamente usando primitivas de sistema de

ficheros. Como tal, el BA podría no ser un miembro de los administradores de SQL en el aspecto alternativo.

El resultado de red de la autenticación es un testigo de seguridad que representa el principal que evalúa el sistema 400 de ficheros. Esta estructura de datos puede incluir el SID del principal entrante así como los SID de todos los grupos para los que el principal es un miembro. Además, todos los privilegios mantenidos por el usuario pueden activarse, por defecto, mientras se conecta al sistema 400 de ficheros. Como se entenderá mejor siguiendo el análisis a continuación, este testigo puede usarse posteriormente para realizar decisiones de autorización.

Volviendo ahora a un análisis de autorización, como se ha descrito anteriormente, la autorización del sistema de ficheros puede integrarse en seguridad de nivel de compartición y seguridad de nivel de elemento. Como se usa en esta descripción, una “compartición” puede hacer referencia a un alias a un elemento 408 en el almacenamiento 410. Cuando se crea un almacenamiento 410, se crea una compartición por defecto con alias al elemento raíz. Los usuarios con suficientes privilegios pueden crear comparticiones con alias a cualquier contenedor genérico (por ejemplo, el elemento 408) en el almacenamiento 410.

El sistema de ficheros puede usar rutas por convenio de nomenclatura para exponer el espacio de nombres local y remotamente. Por lo tanto los clientes del sistema de ficheros se conectan a una compartición mediante la cual el punto de conexión junto con la jerarquía relativa de nombres constituye el mecanismo de direccionamiento a los objetos 408 del sistema de ficheros.

A modo de ejemplo, supóngase que un usuario se conecta a una compartición raíz para acceder a foo. Por consiguiente, el acceso aparecería como \\MachineName\StoreName\RootShare\...\foo. De manera similar, el usuario conectado a una compartición denominada AliceShare accedería al mismo objeto que \\MachineName\AliceShare\...\foo. En este ejemplo, el permiso efectivo en el elemento puede ser una función del descriptor de seguridad en la compartición conectada y el elemento. Se ha de entender que lo primero define una seguridad de nivel de compartición y lo último define una seguridad de nivel de elemento. Los detalles sobre cada uno de estos mecanismos de seguridad así como reglas para composición de los descriptores de seguridad efectivos se describen a continuación.

Comenzando con un análisis de la seguridad de nivel de compartición, las comparticiones del sistema de ficheros de acuerdo con la invención son algo parecidas a las comparticiones de estilo Windows. Para proporcionar semánticas uniformes a través de acceso local y remoto, para cada compartición de sistema de ficheros creada, puede crearse también una compartición en espejo. Las comparticiones pueden almacenarse como elementos en un almacenamiento de catálogo y pueden asegurarse usando seguridad de elemento que es el asunto que sigue. Los permisos en estos elementos y en las comparticiones pueden ser la misma semántica de acceso uniforme de concesión tanto en acceso local como remoto.

Los permisos por defecto pueden concederse según se desee con respecto a los elementos. Por ejemplo, elementos distintos en una compartición pueden tener diferentes permisos por defecto aplicados con respecto a unas características de usuario (por ejemplo, administrador integrado de sistema local, autenticado, interactivo...).

Similar a las comparticiones de estilo Windows, los valores por defecto para el descriptor de seguridad de compartición son configurables usando el ajuste de registro en LanManServer\DefaultSecurity\SrvsvcDefaultShareInfo.

Los mecanismos de seguridad de elemento pueden emplear descriptores de seguridad para efectuar control de acceso. Por consiguiente, en un aspecto, un descriptor de seguridad puede comunicarse mediante las API (interfaces de programa de aplicación) en un formato de cadena de lenguaje de definición de descriptor de seguridad y almacenarse en la base de datos en un formato binario empaquetado bajo la columna VARBINARY de Sys.Security_Descriptors, la tabla de descriptor de seguridad (202 de la Figura 2).

Existe una nueva tabla de descriptor de seguridad, 202 de la Figura 2 como ha descrito anteriormente, Sys.Security_Descriptors, para mantener cada descriptor de seguridad único, almacenarse como un descriptor de seguridad binario empaquetado con un ID único (SDID) para uso como una clave ajena en las tablas de base del sistema de ficheros. Por ejemplo, una tabla de descriptor de seguridad puede parecer como sigue:

SDID	SecurityDescriptor VARBINARY
55	XXXXXXXXXX
56	XXXXXXXXXX

Aunque la tabla de descriptor de seguridad anterior emplea una representación binaria para el descriptor de seguridad, se ha de apreciar que puede emplearse cualquier representación adecuada sin alejarse del alcance de la invención según se reivindica.

Haciendo referencia ahora a un análisis de representación y almacenamiento de descriptores de seguridad y datos relacionados, como se ha descrito anteriormente, la invención emplea dos tablas internas que pueden mantener información relacionada con el descriptor de seguridad - una tabla de descriptor de seguridad (por ejemplo, sys.security_descriptors y una tabla de instancia única (por ejemplo, [System.Storage.Store].[Table!SecurityDescriptorSingleInstance]).

Continuando con el ejemplo, Sys.security_descriptors es una vista de catálogo mantenida por SQL. Este binario se almacena en una fila correspondiente con el SDID.

La tabla de instancia única puede mantenerse mediante el sistema de ficheros. Contiene un mapa de un hash del descriptor de seguridad binario al SDID identificado en la vista o tabla Sys.security_descriptors anteriormente mencionada. En un ejemplo, puede emplearse un hash SHA-1. En un aspecto, si se crean múltiples elementos con los mismos descriptores de seguridad, puede existir una única entrada en ambas de las tablas.

Como se ha establecido anteriormente, otra característica novedosa de la invención es que si la tabla de instancia única estuviera corrupta alguna vez, puede destruirse ya que es una tabla de auto-regeneración. En otras palabras, si tuviera lugar una corrupción, puede crearse una nueva tabla simplemente generando nuevos valores hash y asociándolos al SDID apropiado.

En un aspecto, las tablas Elemento/Extensión/Fragmento/Enlace pueden tener una entrada para el SDID que se marca con el atributo “seguridad”. Se entenderá que esto puede asegurar que cualquier acceso de lectura a estas tablas y cualquier vista creada en la parte superior de estas vistas podría someterse a una comprobación de acceso que solicite (FILE_READ_DATA | FILE_READ_ATTRIBUTES). Se entenderá adicionalmente que la tabla Extensión de elemento, enlace y Fragmento de elemento debe tener la misma tabla de descriptor de seguridad que la tabla Elemento.

La Figura 5 ilustra una metodología de inicialización de acuerdo con un aspecto de la invención. Aunque, para fines de simplicidad de explicación, se muestra la una o más metodologías mostradas en el presente documento, por ejemplo, en forma de un diagrama de flujo y se describen como una serie de actos, se ha de entender y apreciar que la invención objeto no está limitada por el orden de los actos, ya que algunos actos pueden tener lugar, de acuerdo con la invención, en un orden diferente y/o de manera concurrente con otros actos de los que se muestran y describen en el presente documento. Por ejemplo, los expertos en la materia entenderán y apreciarán que una metodología podría representarse, como alternativa, como una serie de estados o eventos interrelacionados, tales como en un diagrama de estado. Además, no todos los actos ilustrados pueden requerirse para implementar una metodología de acuerdo con la invención.

Mientras se crea una base de datos de modelo durante el procedimiento de creación se inicializan las estructuras de datos de seguridad. En 502, se configuran las tablas. En un ejemplo, configurar las tablas puede configurar Sys.server_principals, Sys.database_principals, Sys.server_role_members y Sys.database_role_members. En 504, se crea una tabla de instancia única. De acuerdo con nuestro ejemplo, [System.Storage.Store]. [Table!SecurityDescriptorSingleInstance] puede crearse en 504.

En 506 se crea un descriptor de seguridad raíz. Este descriptor de seguridad raíz corresponde a la raíz del almacenamiento (por ejemplo, los administradores tienen control total). En 508, se crean los descriptores de elementos de niveles de seguridad. Por ejemplo, en 508, pueden crearse descriptores de seguridad para elementos en collage de manera que los administradores tienen control total y los usuarios autenticados tienen acceso de lectura. En 510, estas entradas se añaden a la tabla de instancia única.

El sistema de ficheros puede soportar heredar las ACL. Por ejemplo, desde el momento de la creación de elemento (por ejemplo, CreateItem o CreateComplexItems), el descriptor de seguridad para el elemento puede calcularse usando el descriptor de seguridad suministrado (si lo hubiera), el descriptor de seguridad padre, el tipo de elemento y el testigo (por ejemplo, testigo de estilo NT) del solicitante.

Haciendo referencia ahora a un análisis de comprobaciones de acceso, todas las API de actualización realizan comprobaciones de acceso apropiadas solicitando [System.Storage.Store].[HasSecurityAccess]. La API asegura que se conceda al solicitante el bit de permiso de solicitud tanto en el nivel de compartición así como en el nivel de descriptor de seguridad (por ejemplo, elemento, registro). En un aspecto específico, la comprobación de acceso realizada en el descriptor de seguridad (del padre) es diferente (FILE_DELETE_CHILD) de la (DELETE) realizada en la compartición. Para otros casos, las dos comprobaciones de acceso pueden ser coherentes.

Continuando con el ejemplo, la propagación de ACL a través de la estructura similar a árbol puede realizarse cuando se solicita SetItemSecurity (con una nueva DACL o SACL) o MoveItem con un nuevo padre. Después de que se realizan las comprobaciones de acceso apropiadas para asegurar que el solicitante está permitido a realizar la operación, puede efectuarse la propagación de ACL en el contexto de sistema de ficheros. No se hacen comprobaciones de acceso en la estructura similar a subárbol para la que se actualizan las ACL.

Se ha de apreciar que la invención puede emplear propagación asíncrona y/o síncrona. Lo siguiente es un análisis de la propagación síncrona. Se ha de entender que la raíz de la estructura similar a subárbol no tiene nada que hacer con elementos compuestos. En su lugar, la raíz de la estructura similar a subárbol es un término genérico para describir el nodo en el que se solicita SetItemSecurity o MoveItem.

De acuerdo con la propagación síncrona, se calcula el nuevo descriptor de seguridad para el elemento de raíz. Si no se actualiza DACL o SACL, el SDID si se actualizó para las tablas elemento, extensión, fragmento y enlace y el sistema retorna. Toda la estructura similar a subárbol de elemento se bloquea en el inicio del elemento. En el ejemplo, no es necesario bloquear ninguna otra tabla (Extensión, Fragmento, Enlace).

A continuación, puede crearse una tabla temporal que contiene todos los elementos en el acto anterior. La tabla temporal puede tener las siguientes características. La tabla temporal puede tener ContainerId, ItemId y NewSdId. Así como, inicialmente, NewSdId puede ser NULO para todos menos la raíz de la estructura similar a subárbol.

Para cada entrada en la tabla temporal, puede calcularse el nuevo SD usando el nuevo SD padre, el tipo del artículo y el SD del artículo existente. En el ejemplo, puede usarse CreatePrivateObjectSecurityEx(SEF_AVOID_PRIVILEGE_CHECK | SEF_AVOID_OWNER_CHECK). Por consiguiente, la tabla temporal puede recorrerse nivel a nivel cada vez que se procesan estas filas cuyo nuevo SD padre se ha calculado y el nuevo SDID para el elemento es NULO. De acuerdo con el ejemplo, esto pasa por la tabla un nivel a la vez.

El número de iteraciones es O (por ejemplo, profundidad de la estructura similar a árbol). Pueden considerarse dos cuestiones. En primer lugar, puede considerarse el cálculo de nuevos descriptores de seguridad. En segundo lugar, puede considerarse la actualización de descriptores de seguridad en todos los hijos. En el segundo escenario, el límite teórico es O (por ejemplo, número de hijos). En el primer escenario, aunque no es necesario, normalmente es O (profundidad del árbol). Si se considera necesario, puede crearse un nuevo descriptor de seguridad (por ejemplo, en las tablas de instancia única y Sys.security_descriptors). A continuación, la tabla de SDID temporal se actualiza en la tabla temporal. Finalmente, puede actualizarse la tabla Elemento, Extensión, Enlace y Fragmento usando los datos calculados en la tabla temporal.

La Figura 6 ilustra que las operaciones T/SQL que consultan las vistas de tabla maestra operan en el contexto de usuario donde se aplica el control de acceso para las sentencias SELECT mediante seguridad de nivel de fila. Adicionalmente, las solicitudes a la API de actualización de almacenamiento de sistema de ficheros se componen en el contexto de usuario pero se ejecutan en el contexto de sistema. La implementación puede aplicar por lo tanto comprobaciones de permiso para el solicitante.

La Figura 7 ilustra un sistema 700 que emplea inteligencia artificial (AI) que facilita automatizar una o más características de acuerdo con la invención objeto. La invención objeto (por ejemplo, en relación con implementar políticas de seguridad) puede emplear diversos esquemas basados en AI para llevar a cabo diversos aspectos de los mismos. Por ejemplo, un procedimiento para determinar si debería establecerse un descriptor de seguridad y, si es así, el nivel de seguridad a emplear mediante un sistema y procedimiento de clasificador automático. Además, cuando las tablas de instancia única y descriptor de seguridad (202, 204 de la Figura 2) están localizadas remotamente en múltiples localizaciones, el clasificador puede emplearse para determinar qué localización se seleccionará para comparación.

Un clasificador es una función que mapea un vector de atributo de entrada, x = (x1, x2, x3, x4, *xn*), a una confianza que la entrada pertenece a una clase, es decir, f(x) = *confianza*(*clase*). Tal clasificación puede emplear un análisis probabilístico y/o estadístico (por ejemplo, factorizar las utilidades y costes del análisis) para pronosticar o inferir una acción que un usuario desea que se realice automáticamente.

Una máquina de vector de soporte (SVM) es un ejemplo de un clasificador que puede emplearse. La SVM opera hallando una híper-superficie en el espacio de posibles entradas, híper-superficie que intenta dividir los criterios de activación a partir de los eventos de no activación. De manera intuitiva, esto hace la clasificación correcta para probar datos que están cerca, pero no idénticos a los datos de entrenamiento. Otros enfoques de clasificación de modelo dirigidos y no dirigidos incluyen, por ejemplo, Bayes simple, redes de Bayes, árboles de decisión, redes neuronales, modelos de lógica difusa y modelos de clasificación probabilística que proporcionan diferentes patrones de independencia que pueden emplearse. La clasificación como se usa en el presente documento también es inclusiva de regresión estadística que se utiliza para desarrollar modelos de prioridad.

Como se apreciará fácilmente a partir de la memoria descriptiva objeto, la invención objeto puede emplear clasificadores que se entrenan explícitamente (por ejemplo, mediante unos datos de entrenamiento genéricos) así como que se entrenan implícitamente (por ejemplo, mediante la observación del comportamiento del usuario, recibir información extrínseca). Por ejemplo, las SVM están configuradas mediante una fase de aprendizaje o entrenamiento en un constructor clasificador y módulo de selección de característica. Por lo tanto, el clasificador o los clasificadores pueden usarse para aprender y realizar automáticamente un número de funciones, incluyendo pero sin limitación, determinación de acuerdo con unos criterios predeterminados.

Haciendo referencia ahora a la Figura 8, se ilustra un diagrama de bloques de un ordenador operable para ejecutar la arquitectura desvelada. Para proporcionar contexto adicional para diversos aspectos de la invención objeto, la Figura 8 y el siguiente análisis se pretenden para proporcionar una descripción general, breve, de un entorno 800 informático adecuado en el que pueden implementarse los diversos aspectos de la invención. Aunque la invención se ha descrito anteriormente en el contexto general de instrucciones ejecutables por ordenador que pueden ejecutarse en uno o más componentes, los expertos en la materia reconocerán que la invención también puede implementarse en combinación con otros módulos de programa y/o como una combinación de hardware y software.

En general, módulos de programa incluyen rutinas, programas, componentes, estructuras de datos, etc., que realizan tareas particulares o implementan tipos de datos abstractos particulares. Además, los expertos en la materia apreciarán que los procedimientos inventivos pueden ponerse en práctica con otras configuraciones de sistema informático, incluyendo sistemas informáticos de único procesador o multiprocesador, miniordenadores, ordenadores centrales, así como ordenadores personales, dispositivos informáticos portátiles, electrónica basada en microprocesador o de consumo programable, y similares, cada uno de los cuales puede acoplarse de manera operativa a uno o más dispositivos asociados.

Los aspectos ilustrados de la invención pueden ponerse en práctica también en entornos informáticos distribuidos donde ciertas tareas se realizan mediante dispositivos de procesamiento remoto que están enlazados a través de una red de comunicaciones. En un entorno informático distribuido, pueden localizarse módulos de programa tanto en dispositivos de almacenamiento de memoria locales como remotos.

Un ordenador típicamente incluye una diversidad de medios legibles por ordenador. Medio legible por ordenador puede ser cualquier medio disponible que pueda accederse por el ordenador e incluye tanto medio volátil como no volátil, memoria extraíble y no extraíble. A modo de ejemplo, y no como limitación, medio legible por ordenador puede comprender medio de almacenamiento informático y medio de comunicación. Medio de almacenamiento informático incluye tanto medio volátil como no volátil, extraíble y no extraíble implementado en cualquier procedimiento o tecnología para el almacenamiento de información tal como instrucciones legibles por ordenador, estructuras de datos, módulos de programa u otros datos. Medio de almacenamiento informático incluye, pero sin limitación, RAM, ROM, EEPROM, memoria flash u otra tecnología de memoria, CD-ROM, disco de vídeo digital (DVD) u otro almacenamiento de disco óptico, cartuchos magnéticos, cinta magnética, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, o cualquier otro medio que pueda usarse para almacenar la información deseada y que pueda accederse mediante el ordenador.

El medio de comunicación típicamente incorpora instrucciones legibles por ordenador, estructuras de datos, módulos de programa u otros datos en una señal de datos modulada tal como una onda portadora u otro mecanismo de transporte, e incluye cualquier medio de suministro de información. La expresión “señal de datos modulada” significa una señal que tiene una o más de sus características establecidas o cambiadas de tal manera para codificar información en la señal. A modo de ejemplo, y no como limitación, medio de comunicación incluye medio alámbrico tal como una red alámbrica o conexión de cable directa, y medio inalámbrico tal como medio acústico, de RF, infrarrojos y otros. Deberían incluirse también combinaciones de cualquiera de los anteriores dentro del alcance de medio legible por ordenador.

Con referencia de nuevo a la Figura 8, el entorno 800 ejemplar para implementar diversos aspectos de la invención incluye un ordenador 802, incluyendo el ordenador 802 una unidad 804 de procesamiento, una memoria 806 de sistema y un bus 808 de sistema. El bus 808 de sistema acopla componentes de sistema incluyendo, pero sin limitación, la memoria 806 de sistema a la unidad 804 de procesamiento. La unidad 804 de procesamiento puede ser cualquiera de diversos procesadores comercialmente disponibles. Pueden emplearse también microprocesadores duales y otras arquitecturas de múltiples procesadores como la unidad 804 de procesamiento.

El bus 808 de sistema puede ser cualquiera de varios tipos de estructuras de bus que pueden interconectar adicionalmente a un bus de memoria (con o sin un controlador de memoria), un bus periférico, y un bus local usando cualquiera de una diversidad de arquitecturas de bus comercialmente disponibles. La memoria 806 de sistema incluye memoria de solo lectura (ROM) 810 y memoria de acceso aleatorio (RAM) 812. Un sistema básico de entrada/salida (BIOS) se almacena en una memoria 810 no volátil tal como ROM, EPROM, EEPROM, BIOS que contiene las rutinas básicas que ayudan a transferir información entre elementos en el ordenador 802, tal como durante el arranque. La RAM 812 puede incluir también una RAM de alta velocidad tal como RAM estática para almacenar en caché datos.

El ordenador 802 incluye adicionalmente una unidad interna de disco duro (HDD) 814 (por ejemplo, EIDE, SATA), unidad 814 de disco duro interna que puede configurarse también para uso externo en un chasis adecuado (no mostrado), una unidad de disco magnético flexible (FDD) 816, (por ejemplo, para leer desde o escribir en un disquete 818 extraíble) y una unidad 820 de disco óptico, (por ejemplo, para leer un disco 822 de CD-ROM o, para leer desde o escribir en otro medio óptico de alta capacidad tal como el DVD). La unidad 814 de disco duro, unidad 816 de disco magnético y unidad 820 de disco óptico pueden conectarse al bus 808 de sistema mediante una interfaz 824 de unidad de disco duro, una interfaz 826 de unidad de disco magnético y una 828 interfaz de disco óptico, respectivamente. La interfaz 824 para implementaciones de unidad externa incluye al menos una o ambas de tecnologías de interfaz de Bus Serie Universal (USB) e IEEE 1394. Otras tecnologías de conexión de unidad externa

están dentro de la contemplación de la invención objeto.

Las unidades y sus medios legibles por ordenador asociados proporcionan almacenamiento no volátil de datos, estructuras de datos, instrucciones ejecutables por ordenador, y así sucesivamente. Para el ordenador 802, las unidades y medios acomodan el almacenamiento de cualquier dato en un formato digital adecuado. Aunque la descripción de medio legible por ordenador anterior hace referencia a un HDD, un disquete magnético extraíble y un medio óptico extraíble tal como un CD o DVD, debería apreciarse por los expertos en la materia que otros tipos de medios que son legibles por un ordenador, tal como unidades zip, cintas magnéticas, tarjetas de memoria flash, cartuchos y similares, pueden usarse también en el entorno de operación ejemplar, y además, cualquier medio de este tipo puede contener instrucciones ejecutables por ordenador para realizar los procedimientos de la invención.

Puede almacenarse un número de módulos de programa en las unidades y RAM 812, incluyendo un sistema 830 operativo, uno o más programas 832 de aplicación, otros módulos 834 de programa y datos 836 de programa. Todo o porciones del sistema operativo, aplicaciones, módulos y/o datos pueden almacenarse también en caché en la RAM 812. Se aprecia que la invención puede implementarse con diversos sistemas operativos o combinaciones de sistemas operativos comercialmente disponibles.

Un usuario puede introducir comandos e información en el ordenador 802 a través de uno o más dispositivos de entrada alámbricos/inalámbricos, por ejemplo un teclado 838 y un dispositivo apuntador, tal como un ratón 840. Otros dispositivos de entrada (no mostrados) pueden incluir un micrófono, un control remoto de IR, una palanca de mandos, un control de juegos, un lápiz óptico, pantalla táctil o similares. Estos y otros dispositivos de entrada a menudo están conectados a la unidad 804 de procesamiento a través de una interfaz 842 de dispositivo de entrada que está acoplada al bus 808 de sistema, pero que puede conectarse mediante otras interfaces, tales como un puerto paralelo, un puerto serie IEEE 1394, un puerto de juegos, un puerto USB, una interfaz de IR, etc.

Un monitor 844 u otro tipo de dispositivo de visualización está también conectado al bus 808 de sistema mediante una interfaz, tal como un adaptador 846 de vídeo. Además del monitor 844, un ordenador típicamente incluye otros dispositivos de salida periféricos (no mostrados), tales como altavoces, impresoras, etc.

El ordenador 802 puede operar en un entorno en red usando conexiones lógicas mediante comunicaciones alámbricas y/o inalámbricas a uno o más ordenadores remotos, tales como un ordenador u ordenadores 848 remotos. El ordenador u ordenadores 848 remotos pueden ser una estación de trabajo, un ordenador de servidor, un encaminador, un ordenador personal, ordenador portátil, aparato de entretenimiento basado en microprocesador, un dispositivo de pares u otro nodo de red común, y típicamente incluyen muchos o todos los elementos descritos con relación al ordenador 802, aunque, por fines de brevedad, únicamente se ilustra un dispositivo 850 de memoria/almacenamiento. Las conexiones lógicas representadas incluyen conectividad alámbrica/inalámbrica a una red de área local (LAN) 852 y/o redes mayores, por ejemplo, una red de área extensa (WAN) 854. Tales entornos de red LAN y WAN son lugares comunes en oficinas y empresas, y facilitan las redes informáticas a nivel de empresa, tales como intranets, todas las cuales pueden conectarse a una red de comunicaciones global, por ejemplo, internet.

Cuando se usa en un entorno de interconexión en LAN, el ordenador 802 está conectado a la red 852 local a través de una interfaz o adaptador 856 de red de comunicación alámbrica y/o inalámbrica. El adaptador 856 puede facilitar la comunicación alámbrica o inalámbrica a la LAN 852, que puede incluir también un punto de acceso inalámbrico dispuesto en el mismo para comunicar con el adaptador 856 inalámbrico.

Cuando se usa en un entorno de interconexión en red WAN, el ordenador 802 puede incluir un módem 858, o está conectado a un servidor de comunicaciones en la WAN 854, o tiene otros medios para establecer comunicaciones a través de la WAN 854, tal como por medio de internet. El módem 858, que puede ser interno o externo y un dispositivo alámbrico o inalámbrico, está conectado al bus 808 de sistema mediante la interfaz 842 de puerto serie. En un entorno en red, pueden almacenarse módulos de programa representados con respecto al ordenador 802, o porciones del mismo, en el dispositivo 850 de memoria/almacenamiento remoto. Se apreciará que las conexiones de red mostradas son ejemplares y que pueden usarse otros medios de establecimiento de un enlace de comunicaciones entre los ordenadores.

El ordenador 802 es operable para comunicar con cualquier dispositivo inalámbrico o entidades dispuestas de manera operativa en comunicación inalámbrica, por ejemplo, una impresora, escáner, ordenador de sobremesa y/o portátil, asistente de datos portátil, comunicaciones por satélite, cualquier pieza de equipo o localización asociada con una etiqueta detectable de manera inalámbrica (por ejemplo, un quiosco, puesto de periódicos, servicios), y teléfono. Esto incluye al menos tecnologías inalámbricas Wi-Fi y Bluetooth™. Por lo tanto, la comunicación puede ser una estructura predefinida como con una red convencional o simplemente una comunicación ad hoc entre al menos dos dispositivos.

Wi-Fi, o Fidelidad Inalámbrica, permite la conexión a internet desde un sofá en casa, una cama en una habitación de hotel, o una sala de conferencias en el trabajo, sin cables. Wi-Fi es una tecnología inalámbrica similar a la usada en un teléfono celular que posibilita a tales dispositivos, por ejemplo, ordenadores, enviar y recibir datos hacia el interior y exterior; en cualquier lugar dentro del alcance de una estación base. Las redes Wi-Fi usan tecnologías de radio denominadas IEEE 802.11 (a, b, g, etc.) para proporcionar conectividad inalámbrica segura, fiable, rápida. Una red

Wi-Fi puede usarse para conectar ordenadores entre sí, a internet, y a redes alámbricas (que usan IEEE 802.3 o Ethernet). Las redes Wi-Fi operan en las bandas de radio de 2,4 y 5 GHz sin licencia a una velocidad de datos de 11 Mbps (802.11a) o 54 Mbps (802.11b), por ejemplo, o con productos que contienen ambas bandas (banda dual), de modo que las redes pueden proporcionar rendimiento en el mundo real similar a las redes de Ethernet 10BaseT básicas usadas en muchas oficinas.

Haciendo referencia ahora a la Figura 9, se ilustra un diagrama de bloques esquemático de un entorno 900 informático ejemplar de acuerdo con la invención objeto. El sistema 900 incluye uno o más cliente o clientes 902. El cliente o clientes 902 pueden ser hardware y/o software (por ejemplo, hilos, procedimientos, dispositivos informáticos). El cliente o clientes 902 pueden alojar la cookie o cookies y/o información contextual asociada empleando la invención, por ejemplo.

El sistema 900 también incluye uno o más servidor o servidores 904. El servidor o servidores 904 pueden ser también hardware y/o software (por ejemplo, hilos, procedimientos, dispositivos informáticos). Los servidores 904 pueden alojar hilos para realizar transformaciones empleando la invención, por ejemplo. Una comunicación posible entre un cliente 902 y un servidor 904 puede ser en forma de un paquete de datos adaptado para transmitirse entre dos o más procedimientos informáticos. El paquete de datos puede incluir una cookie y/o información contextual asociada, por ejemplo. El sistema 900 incluye una estructura 906 de comunicación (por ejemplo, una red de comunicación global tal como internet) que puede emplearse para facilitar las comunicaciones entre el cliente o clientes 902 y el servidor o servidores 904.

Las comunicaciones pueden facilitarse mediante una tecnología alámbrica (incluyendo fibra óptica) y/o inalámbrica. El cliente o clientes 902 están conectados de manera operativa a uno o más almacenamiento o almacenamientos 908 de datos de cliente que pueden emplearse para almacenar información local al cliente o clientes 902 (por ejemplo, cookie o cookies y/o información contextual asociada). De manera similar, el servidor o servidores 904 están conectados de manera operativa a uno o más almacenamiento o almacenamientos 910 de datos de servidor que pueden emplearse para almacenar información local a los servidores 904. Lo que se ha descrito anteriormente incluye ejemplos de la invención. Por supuesto, no es posible describir cada combinación concebible de componentes o metodologías para fines de descripción de la invención objeto, pero un experto en la materia puede reconocer que son posibles combinaciones y permutaciones adicionales de la invención. Por consiguiente, la invención se pretende abarcar todas tales alteraciones, modificaciones y variaciones que caen dentro del alcance de las reivindicaciones adjuntas. Adicionalmente, hasta el punto que el término “incluye” se usa en cualquiera de la descripción detallada o las reivindicaciones, tal término se pretende que sea inclusivo en una manera similar a la expresión “que comprende” ya que “que comprende” se interpreta cuando se emplea como una expresión transicional en una reivindicación.

REIVINDICACIONES

1. Un sistema informático que facilita acceder a datos almacenados de acuerdo con una representación jerárquica, en el que una política de seguridad está asociada a cada dato o contenedor de datos en la representación jerárquica, que comprende:

un componente (102) de consulta adaptado para consultar los datos almacenados para proporcionar una agregación de datos que abarca múltiples jerarquías de contenedor con políticas de seguridad heterogéneas; y
un componente (104) de seguridad de nivel de fila adaptado para filtrar la agregación de datos basándose en al menos un permiso de acceso de nivel de fila de un principal para proporcionar un conjunto de datos resultante que son todos los datos a través de toda la representación jerárquica para la que el al menos un permiso de acceso de nivel de fila del principal satisface la política de seguridad de los datos.

2. El sistema de la reivindicación 1, que comprende adicionalmente un componente que proporciona un sistema de establecimiento de identidad confiable usado en relación con una política de aplicación de control de acceso.

3. El sistema de la reivindicación 1, que comprende adicionalmente un componente de representación que representa la abstracción limitada.

4. El sistema de la reivindicación 1, donde el componente de seguridad de nivel de fila asocia una política de seguridad con al menos una fila en el almacenamiento de datos.

5. El sistema de la reivindicación 4, donde cada fila en el almacenamiento de datos contiene un único objeto.

6. El sistema de la reivindicación 5, donde la política de seguridad es al menos uno de una lista de control de acceso, ACL y un descriptor de seguridad.

7. El sistema de la reivindicación 6, donde el objeto es al menos uno de un elemento de datos y un contenedor organizado en una organización jerárquica.

8. El sistema de la reivindicación 7, que comprende adicionalmente un componente que establece la política de seguridad en una raíz de la jerarquía y propaga la política de seguridad a al menos un hijo en la jerarquía.

9. El sistema de la reivindicación 8, donde el componente que propaga la política de seguridad usa un descriptor de seguridad de un padre y el objeto para calcular un descriptor de seguridad efectivo para el objeto.

10. El sistema de la reivindicación 1, donde el componente de seguridad de nivel de fila comprende además:

una tabla de descriptor de seguridad que mapea un descriptor de seguridad a un identificador de descriptor de seguridad, SDID; y
una tabla de instancia única que mapea el SDID a un valor hash del SDID.

11. El sistema de la reivindicación 10, donde el SDID es un valor entero que apunta al descriptor de seguridad.

12. El sistema de la reivindicación 10, donde el valor hash se genera mediante un algoritmo hash SHA-1.

13. El sistema de la reivindicación 1, que comprende adicionalmente un componente de inteligencia artificial, Al, que emplea un análisis probabilístico y/o estadístico para pronosticar o inferir una acción de un usuario.

14. Un procedimiento implementado por ordenador para proporcionar control de acceso a datos almacenados en un almacenamiento de datos, que comprende:

organizar los datos en una organización jerárquica;
acceder a los datos en la organización jerárquica;
establecer una política de seguridad en una raíz de la organización jerárquica;
propagar la política de seguridad a al menos un hijo en la organización jerárquica basándose al menos en parte en un descriptor de seguridad padre;
consultar los datos almacenados para proporcionar una agregación de datos que abarca múltiples jerarquías de contenedor con políticas de seguridad heterogéneas; y
aplicar una política de seguridad de nivel de fila para filtrar la agregación de datos basándose en al menos un permiso de acceso de nivel de fila de un principal; y
proporcionar un conjunto de datos resultante que son todos los datos a través de toda la representación jerárquica para la que el al menos un permiso de acceso de nivel de fila del principal satisfaga la política de seguridad de los datos.

15. El procedimiento de la reivindicación 14, en el que la política de seguridad de nivel de fila asocia al menos uno de una lista de control de acceso, ACL y un descriptor de seguridad con al menos una fila en el almacenamiento de datos.

16. El procedimiento de la reivindicación 14, que comprende adicionalmente establecer un sistema de establecimiento de identidad confiable usado en relación con aplicar la política de seguridad de nivel de fila.

17. El procedimiento de la reivindicación 16, que comprende adicionalmente representar la abstracción limitada.

18. Un medio legible por ordenador que tiene almacenado en el mismo instrucciones ejecutables por ordenador para llevar a cabo el procedimiento de una de las reivindicaciones 14 a 17.

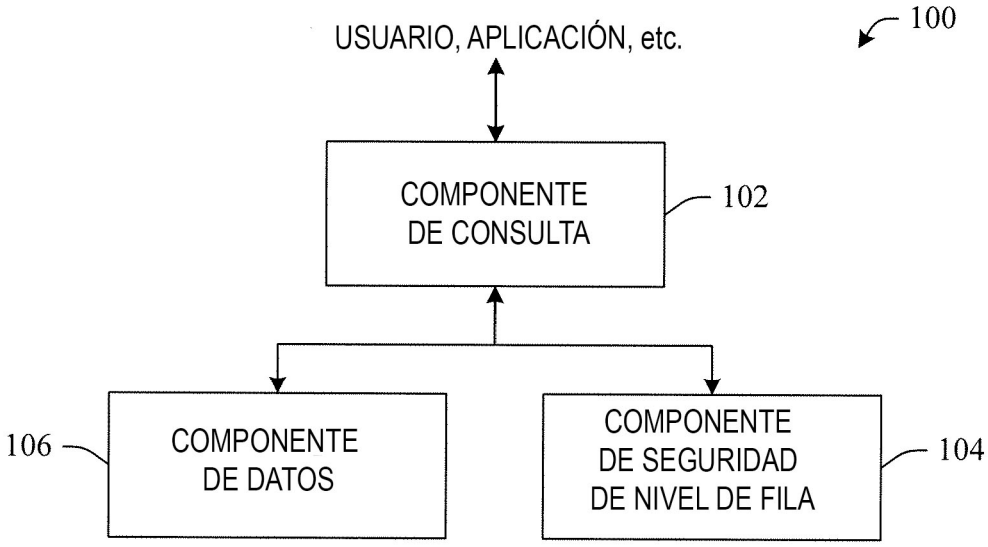
USUARIO, APLICACIÓN, etc.
100
COMPONENTE DE CONSULTA
102
106
COMPONENTE DE DATOS
COMPONENTE DE SEGURIDAD DE NIVEL DE FILA
104

FIG. 1

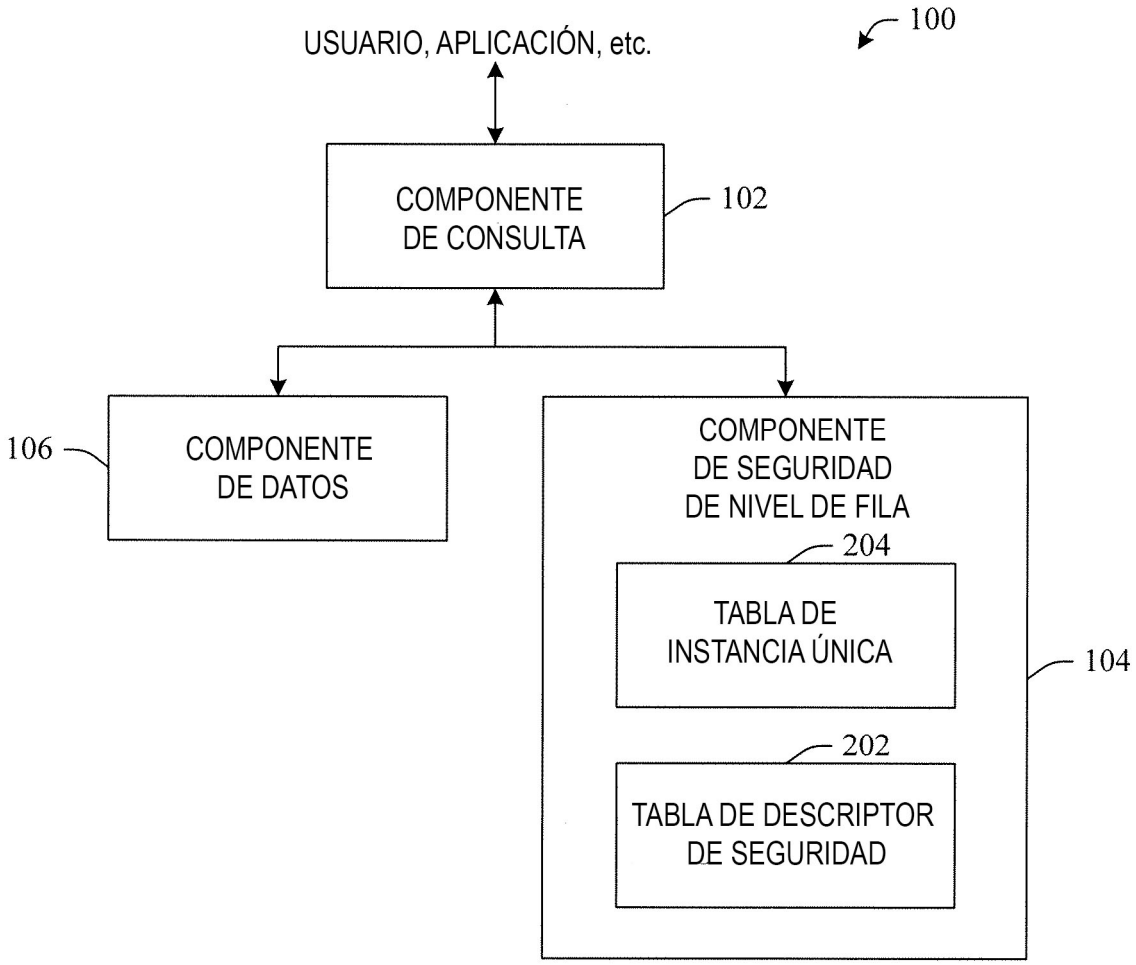
100
USUARIO, APLICACIÓN, etc.
COMPONENTE DE CONSULTA
102
106
COMPONENTE DE DATOS
COMPONENTE DE SEGURIDAD DE NIVEL DE FILA
204
TABLA DE INSTANCIA ÚNICA
104
202
TABLA DE DESCRIPTOR DE SEGURIDAD

FIG. 2

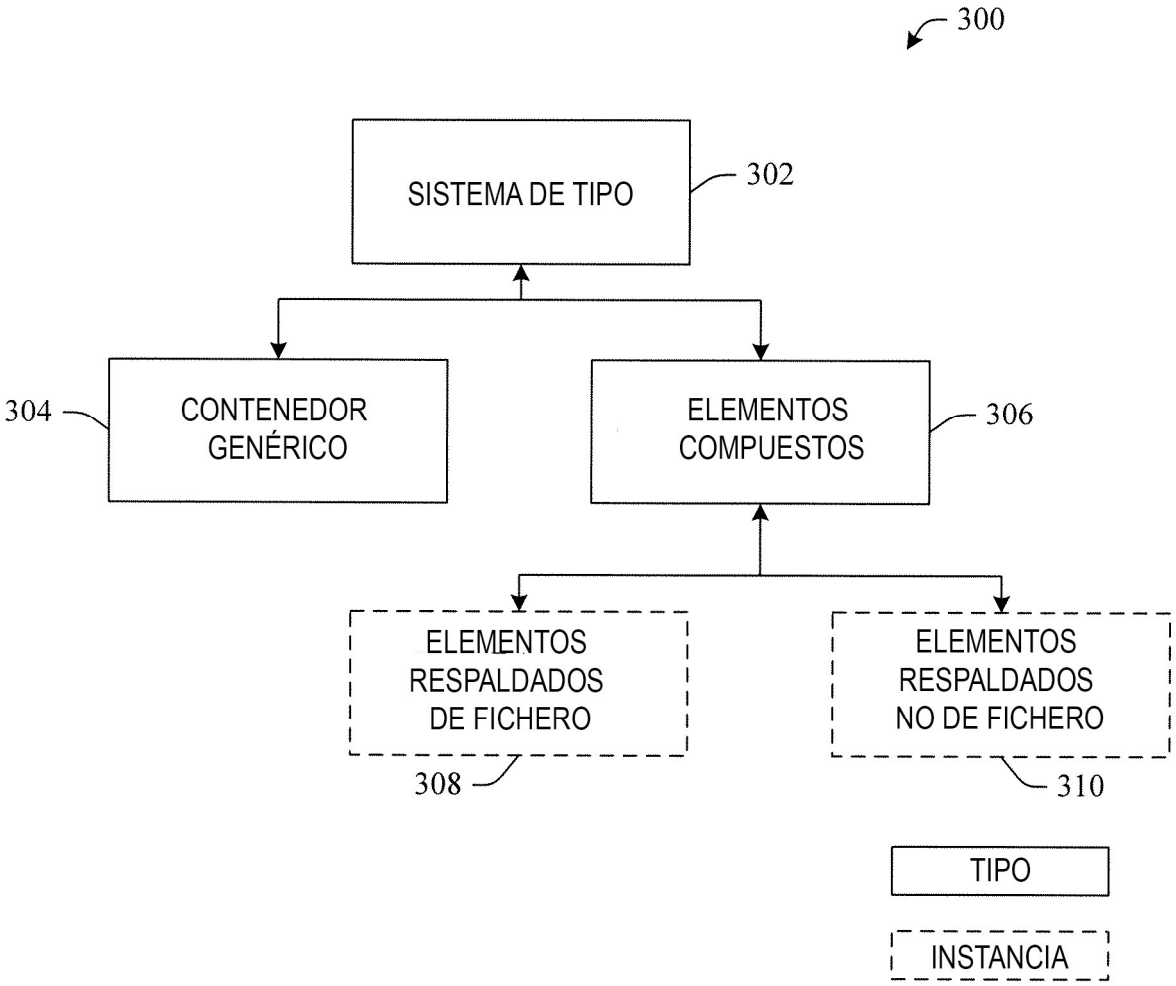
300
SISTEMA DE TIPO
302
304
CONTENEDOR GENÉRICO
ELEMENTOS COMPUESTOS
306
ELEMENTOS RESPALDADOS DE FICHERO
308
ELEMENTOS RESPALDADOS NO DE FICHERO
310
TIPO
INSTANCIA

FIG. 3

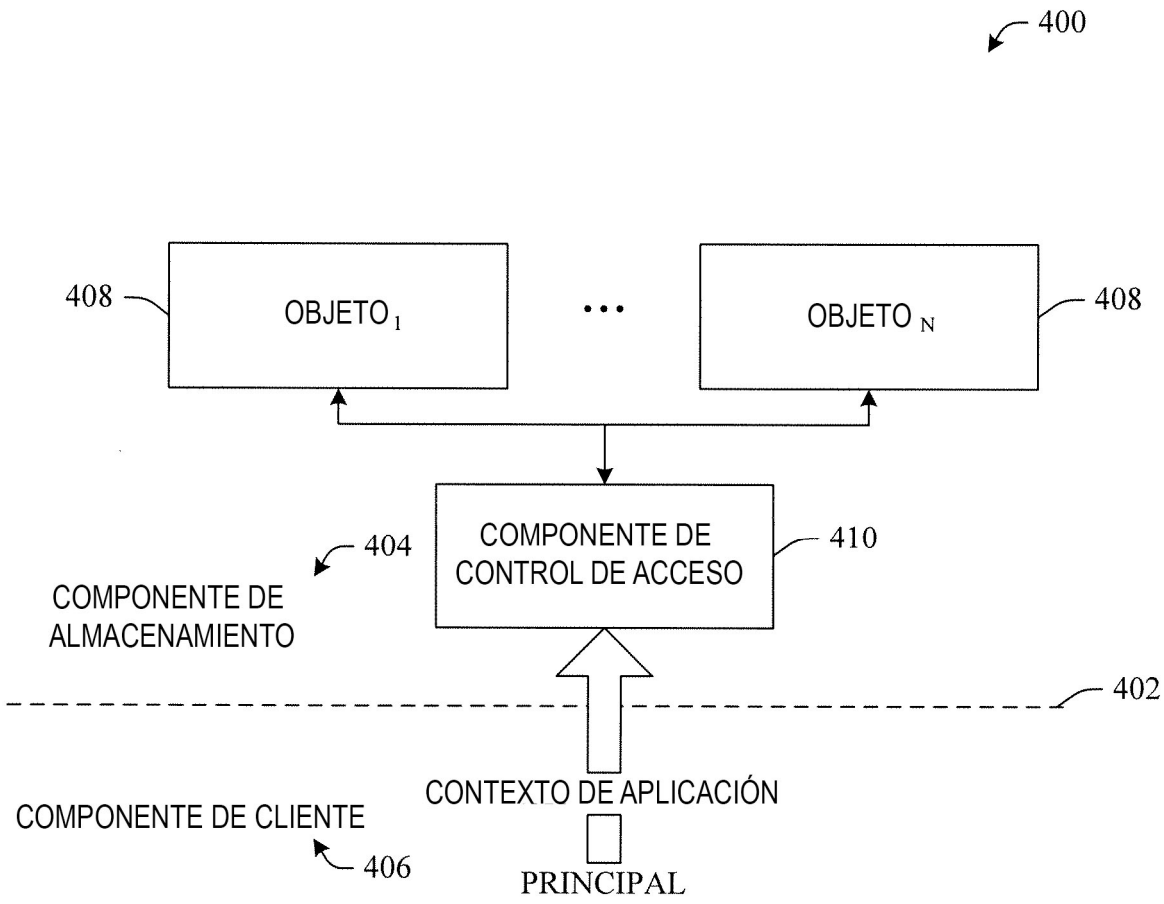
400
408
OBJETO 1
OBJETO N
408
404
COMPONENTE DE ALMACENAMIENTO
COMPONENTE DE CONTROL DE ACCESO
410
402
CONTEXTO DE APLICACIÓN
COMPONENTE DE CLIENTE
406
PRINCIPAL

FIG. 4

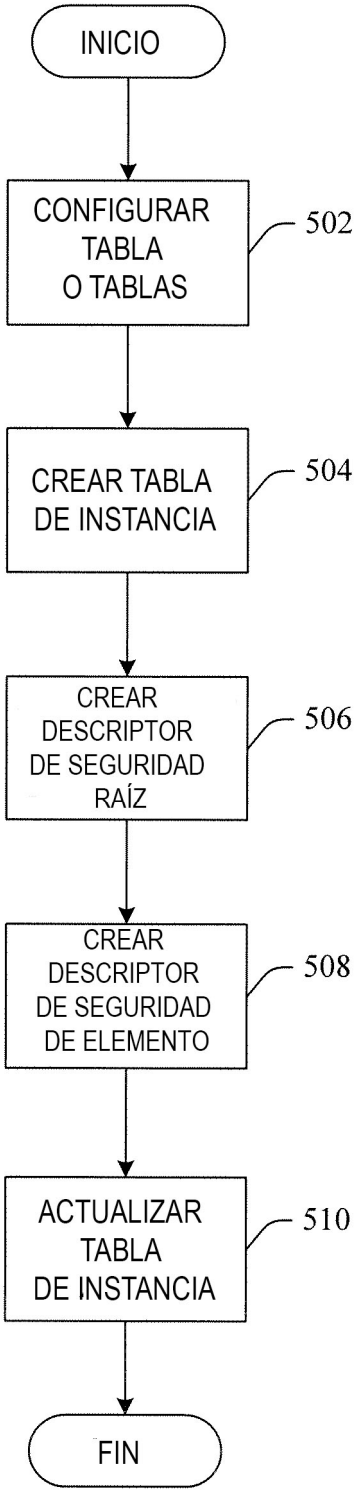
INICIO
CONFIGURAR TABLA O TABLAS
502
CREAR TABLA DE INSTANCIA
504
CREAR DESCRIPTOR DE SEGURIDAD RAÍZ
506
CREAR DESCRIPTOR DE SEGURIDAD DE ELEMENTO
508
ACTUALIZAR TABLA DE INSTANCIA
510
FIN

FIG. 5

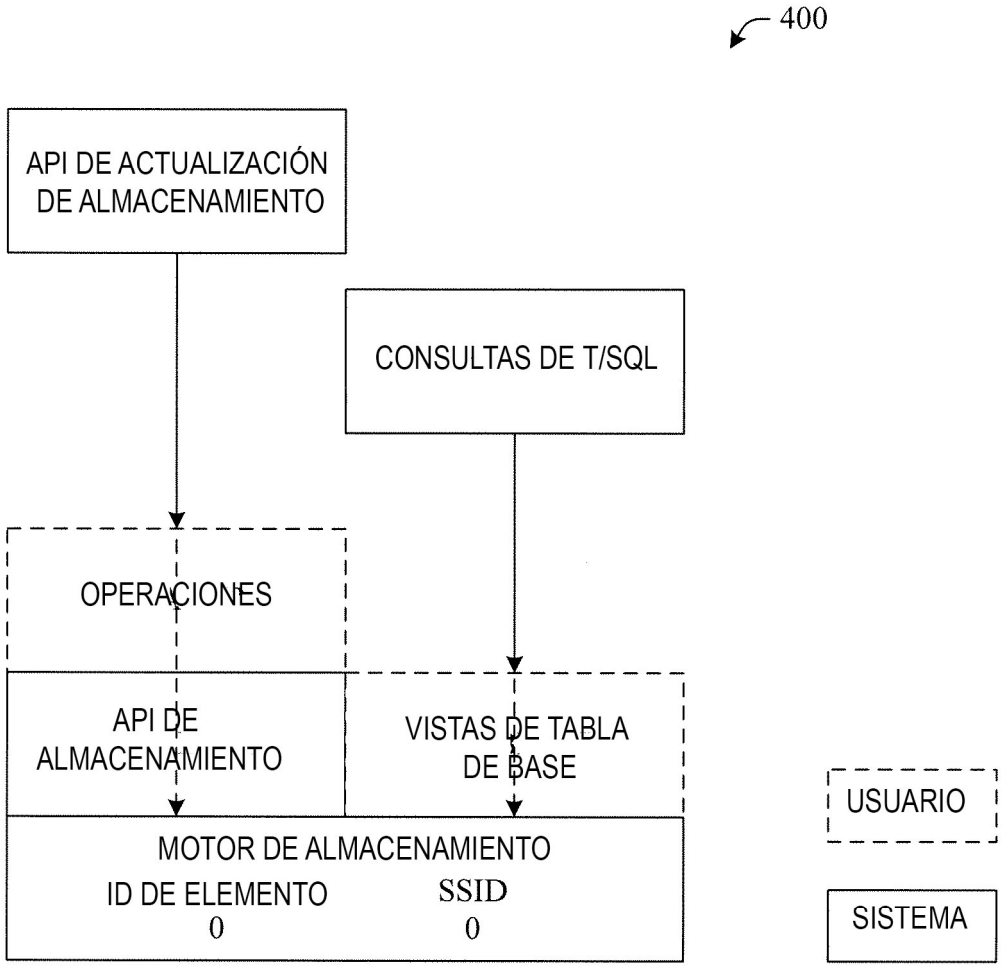
400
API DE ACTUALIZACIÓN DE ALMACENAMIENTO
CONSULTAS DE T/SQL
OPERACIONES
API DE ALMACENAMIENTO
VISTAS DE TABLA DE BASE
MOTOR DE ALMACENAMIENTO
ID DE ELEMENTO
0
SSID
0
USUARIO
SISTEMA

FIG. 6

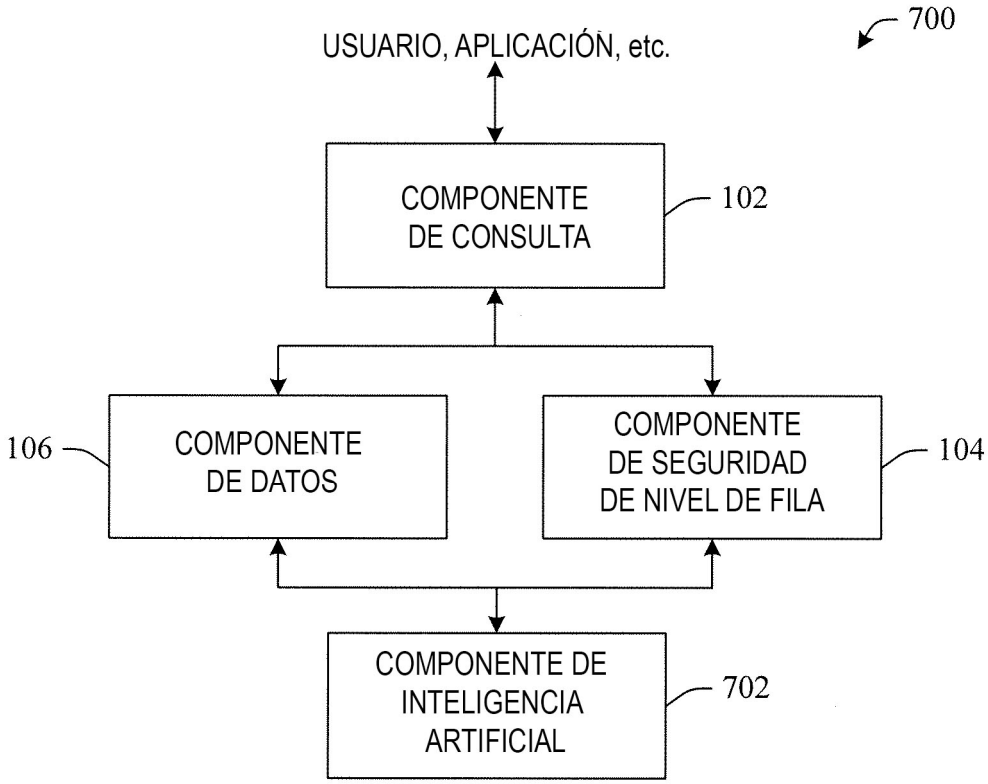
700
USUARIO, APLICACIÓN, etc.
COMPONENTE DE CONSULTA
102
106
COMPONENTE DE DATOS
COMPONENTE DE SEGURIDAD DE NIVEL DE FILA
104
COMPONENTE DE INTELIGENCIA ARTIFICIAL
702

FIG. 7

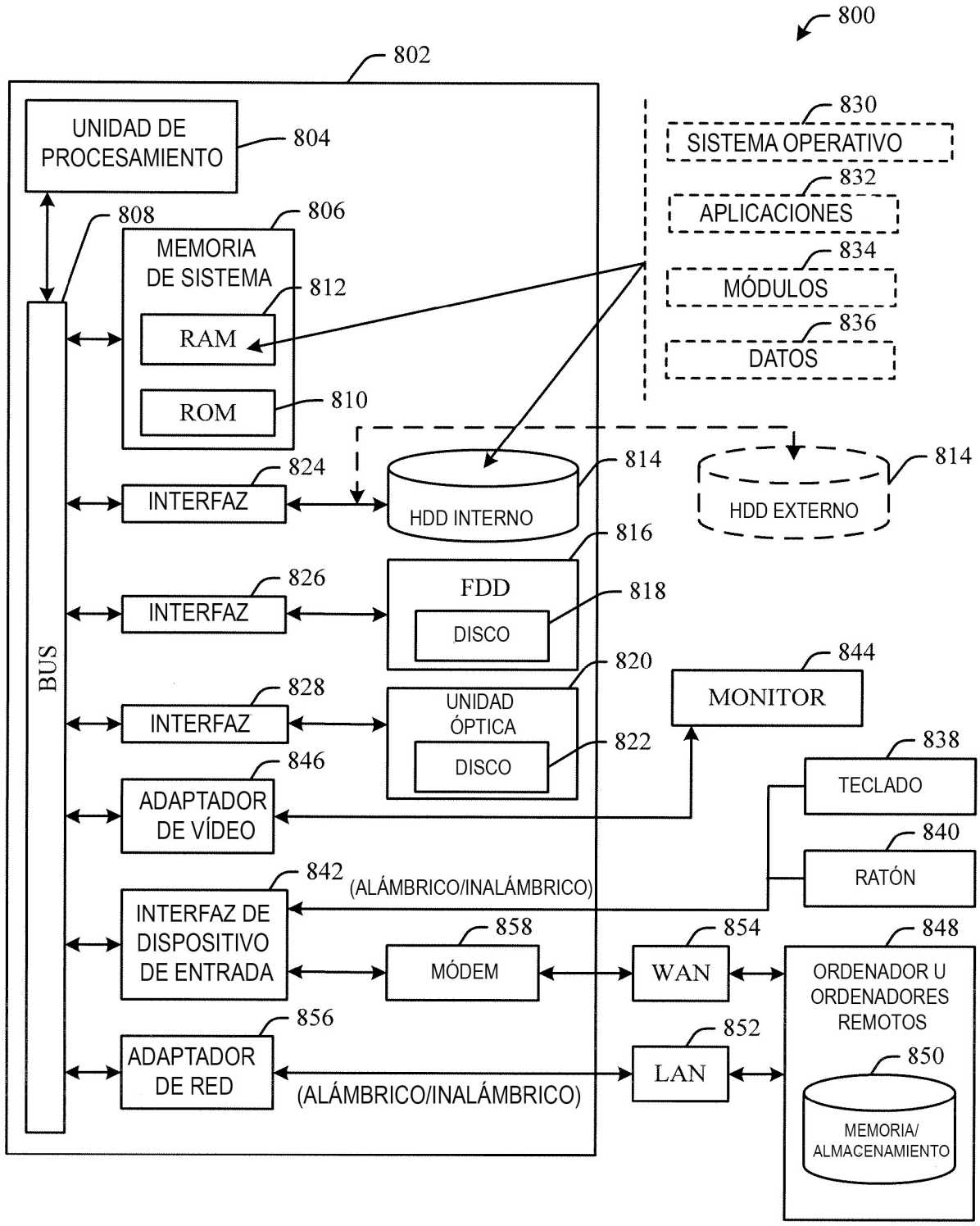
800
802
UNIDAD DE PROCESAMIENTO
804
808
806
MEMORIA DE SISTEMA
812
RAM
ROM
810
BUS
830
SISTEMA OPERATIVO
832
APLICACIONES
834
MÓDULOS
836
DATOS
824
INTERFAZ
HDD INTERNO
814
HDD EXTERNO
814
826
INTERFAZ
FDD
DISCO
816
818
828
INTERFAZ
UNIDAD ÓPTICA
DISCO
820
822
844
MONITOR
838
TECLADO
840
RATÓN
846
ADAPTADOR DE VÍDEO
842
(ALÁMBRICO/INALÁMBRICO)
INTERFAZ DE DISPOSITIVO DE ENTRADA
858
MÓDEM
854
WAN
848
ORDENADOR U ORDENADORES REMOTOS
856
ADAPTADOR DE RED
(ALÁMBRICO/INALÁMBRICO)
852
LAN
850
MEMORIA/ ALMACENAMIENTO

FIG. 8

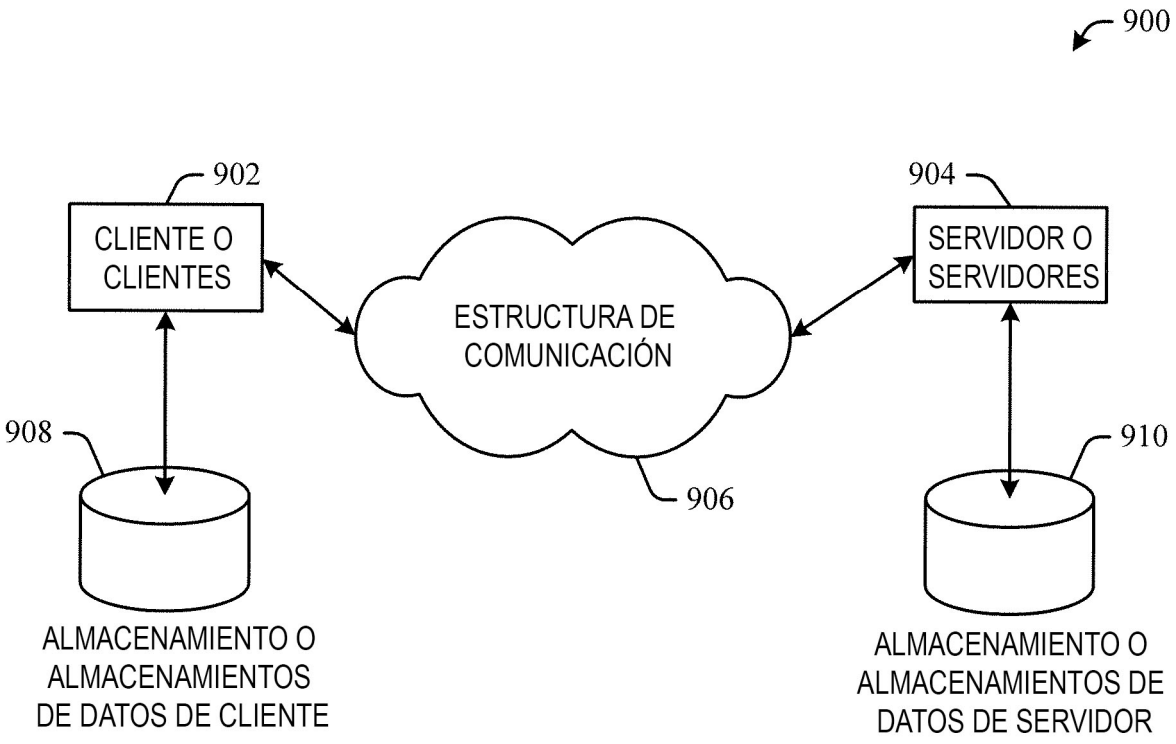
900
902
CLIENTE O CLIENTES
904
SERVIDOR O SERVIDORES
ESTRUCTURA DE COMUNICACIÓN
906
908
910
ALMACENAMIENTO O ALMACENAMIENTOS DE DATOS DE CLIENTE
ALMACENAMIENTO O ALMACENAMIENTOS DE DATOS DE SERVIDOR

FIG. 9