(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 632 740**

(51) Int. CI.:

G06F 9/445 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **16.01.2009 PCT/EP2009/050471**

(87) Fecha y número de publicación internacional: **30.07.2009 WO09092666**

(96) Fecha de presentación y número de la solicitud europea: **16.01.2009 E 09704579 (3)**

(97) Fecha y número de publicación de la concesión europea: **03.05.2017 EP 2235625**

(54) Título: **Método y sistema para desplegar versiones de servidor no compatibles hacia atrás en un entorno informático cliente/servidor**

(30) Prioridad:

22.01.2008 EP 08300042
23.01.2008 US 22834

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
15.09.2017

(73) Titular/es:

AMADEUS S.A.S (100.0%)
485 ROUTE DU PIN MONTARD SOPHIA ANTIPOLIS
06410 BIOT, FR

(72) Inventor/es:

CORDESSES, JOËL;
MONBEL, STÉPHANE;
DOR, PIERRE y
TCHENG, CHRISTOPHE

(74) Agente/Representante:

SUGRAÑES MOLINÉ, Pedro

DESCRIPCIÓN

Método y sistema para desplegar versiones de servidor no compatibles hacia atrás en un entorno informático cliente/servidor

Campo de la invención

La presente invención se refiere, en general, a sistemas informáticos en un entorno cliente/servidor y más específicamente al despliegue de nuevas versiones de aplicaciones de software de servidor que no necesitan ser compatibles con las versiones de cliente.

Antecedentes de la invención

El modelo informático cliente/servidor implementado a través de una red se ha adoptado universalmente. En este modelo, las solicitudes emitidas por las aplicaciones de software de los nodos cliente se envían a uno o más servidores conectados. Una vez procesada por el servidor, la información solicitada se devuelve a los clientes. Este modelo es el de Internet donde los clientes son navegadores web y los servidores son servidores web que incluyen muchos servidores especializados tales como los servidores de correo.

Este modelo es también el de muchos proveedores de servicios que, en general, operan grandes recursos informáticos para servir a una comunidad de clientes, usuarios finales de las aplicaciones de software que se ejecutan en los recursos informáticos del proveedor de servicios, posiblemente sobre una mezcla de redes públicas y privadas incluyendo Internet. Ejemplos de estos proveedores son los sistemas de distribución global (GDS) que proporcionan servicios de viajes a toda la industria de viajes, incluyendo aerolíneas, agencias de viajes tradicionales y en línea (por ejemplo, para planificar y reservar viajes) y aeropuertos (por ejemplo, para el control de salida y el registro de entrada de pasajeros).

Los sistemas de software, en general, se actualizan con frecuencia durante su ciclo de vida. Incluso después de completar la fase de desarrollo, cuando los sistemas están en producción, las aplicaciones de software continúan evolucionando para aplicar correcciones y mejorarlas con la adición de nuevas características. Además, es posible que haya que introducir cambios para aprovechar plenamente las prestaciones de un nuevo tipo de máquina o porque el sistema operativo es nuevo, ha evolucionado o es diferente.

En el modelo cliente/servidor, para desplegar una nueva versión de aplicación de lado del servidor, la práctica convencional es que la nueva versión debe ser compatible hacia atrás. Por lo tanto, cualquiera que sea la versión de la aplicación de cliente que está en uso en cualquier nodo, el nuevo servidor, una vez desplegada, es inmediatamente capaz de manejar las solicitudes del cliente y entregar la información solicitada en un formato compatible.

Este esquema ideal se aplica, en cierta medida, a Internet donde los servidores web necesitan ser compatibles con todos los navegadores web existentes usados por innumerables clientes de la red pública mundial. Esto sin embargo no es completamente cierto. Es bien sabido que no todas las marcas de navegador web reaccionan exactamente de la misma manera y que muchos servidores están soportando en la práctica solo las versiones más recientes de las aplicaciones cliente. Por ejemplo, con Internet Explorer (IE), el navegador web de Microsoft Corporation, el navegador más usado en todo el mundo, muchas aplicaciones de servidor recientes soportan actualmente solo la versión 5 (IE5) y superiores. De hecho, la interfaz gráfica de usuario del cliente (GUI) puede verse gravemente afectada cuando las versiones anteriores del navegador o las marcas de navegador que no se soportan, solicitan y reciben información de un servidor actualizado.

El mantenimiento de la compatibilidad hacia atrás, a pesar de que solo sea parcialmente alcanzable en la práctica, tiene un alto coste. La nueva aplicación de servidor debe hacer frente de una manera u otra a todas las opciones, características, incompatibilidades y defectos de todas las versiones de cliente a soportarse en el campo. Esto tiene un alto coste durante la fase de desarrollo que requiere más tiempo y habilidades para desarrollar la nueva aplicación de servidor y posiblemente requerir más recursos de memoria y de hardware más rápidos para implementarse. Aún más importante, el número de combinaciones de navegadores servidor/cliente a comprobar está creciendo rápidamente hasta un punto donde la fase de prueba puede requerir una cantidad imposible de recursos (máquinas y mano de obra) a poner en marcha para permitir una prueba exhaustiva de todas las combinaciones dentro una cantidad razonable de tiempo.

Lo anterior es cierto a pesar de que cuando se dice que un navegador web es un "cliente delgado", es decir, que un cliente realice actualmente solo una parte limitada del trabajo que debe realizarse entre la aplicación de servidor y la aplicación de cliente que ha solicitado el trabajo realizado. De hecho, la principal tarea de un navegador web es mostrar las páginas devueltas por el servidor web a través de una GUI.

Con los sistemas de cliente/servidor que no están públicamente accesibles, como los GDS mencionados anteriormente, que en general funcionan solo con clientes afiliados (por ejemplo, líneas aéreas, agencias de viajes,

aeropuertos, etc.), la aplicación de cliente puede necesitar más ser un llamado "cliente rico" que tiene que realizar una parte mucho mayor de todo el trabajo a compartir entre el servidor y el cliente. Esto puede ser necesario debido a que el ancho de banda disponible entre los mismos es demasiado limitado para permitir que la aplicación de cliente haga una solicitud al servidor por cada tarea que se va a realizar. Con los GDS esto es, por ejemplo, el caso de los sistemas de aplicación de cliente usados en los aeropuertos para controlar la salida del avión y el registro de entrada de los pasajeros, un trabajo que debe acelerarse en un momento punta cuando los pasajeros están embarcando. De hecho, los aeropuertos pequeños todavía pueden tener recursos limitados de comunicación externa. Además, los recursos informáticos del aeropuerto son propiedad y están bajo el control de las autoridades aeroportuarias que requieren dar su aprobación y por lo tanto tiempo para desplegarse y actualizarse a pesar de que la aplicación de registro de entrada se use por las aerolíneas afiliadas al GDS.

Por lo tanto, cuando la aplicación de cliente es un cliente enriquecido, ideado para realizar por sí mismo una parte más grande del trabajo, el problema de tener servidores compatibles hacia atrás puede incluso ser más difícil, si no imposible, de conseguir que con un cliente ligero. De hecho, el número de opciones y características de todas las versiones de las aplicaciones de cliente enriquecido a soportar es potencialmente mucho mayor, lo que exacerba en gran medida los problemas mencionados anteriormente con respecto a las fases de desarrollo y de prueba de la aplicación de servidor.

El documento US-A-5.732.275 desvela un método y un aparato para gestionar y actualizar automáticamente programas de software. Este documento no aborda el problema de adaptar todas las aplicaciones de software de cliente de una red antes del despliegue de una nueva versión no compatible hacia atrás de una aplicación de software de servidor. De acuerdo con esta técnica anterior, se supone que la aplicación de cliente es capaz por sí misma de descargar una versión de software desde una memoria compartida. Esto restringe el alcance de aplicación de esta tecnología mientras que la presente invención puede aplicarse a una red que incluye terminales bastante tontos con recursos de software limitados.

El documento WO01/69382A desvela un método para la configuración inicial de un dispositivo cliente. De acuerdo con esta publicación, se descarga una nueva plantilla para configurar inicialmente la aplicación de cliente. Esta plantilla está destinada a adaptar la aplicación de cliente a un nuevo formato de organización de datos del servidor y no a una nueva versión de la aplicación de software de servidor en sí.

En vista de lo anterior, es deseable permitir, por lo tanto, en un entorno cliente/servidor, el despliegue de nuevas versiones de aplicaciones de servidor que no necesitan ser compatibles hacia atrás.

Otros objetos, características y ventajas de la presente invención resultarán evidentes para los expertos en la materia tras el examen de la siguiente descripción haciendo referencia a los dibujos adjuntos. Se pretende que cualquier ventaja adicional se incorpore en el presente documento.

Sumario de la invención

Las dificultades mencionadas anteriormente de tener que desplegar servidores compatibles hacia atrás en un entorno cliente/servidor se abordan por la presente invención, que describe un método y un sistema para gestionar la introducción de una nueva versión no compatible hacia atrás de un programa de software de un servidor de aplicaciones en un entorno de red cliente/servidor. El método consiste en distribuir primero a un sistema cliente a servirse por la nueva versión no compatible hacia atrás del programa de software del servidor de aplicaciones, una aplicación de cliente operable en un modo compatible con la versión actual del servidor de aplicaciones y en un modo compatible con la nueva versión no compatible hacia atrás. Tras la instalación en el sistema cliente, la aplicación de cliente se establece en un modo degradado compatible con la versión actual del servidor de aplicaciones. Aunque la versión actual del servidor de aplicaciones está todavía en producción, la aplicación de cliente se mantiene en operación en el modo degradado. Tras la introducción de la nueva versión no compatible hacia atrás del servidor de aplicaciones, la aplicación de cliente se establece en un modo válido compatible con la nueva versión del servidor de aplicaciones. A partir de ese momento, la aplicación de cliente se opera en el modo válido. La configuración de los modos se activa automáticamente en cada relanzamiento del sistema cliente que consulta un servidor de versiones que opera en el entorno de red cliente/servidor. La consulta incluye una identificación del sistema cliente y un número de versión de aplicación de cliente con el fin de obtener del servidor de versiones consultado un valor de estado para operar el sistema cliente en un modo que incluye el modo degradado y el modo válido.

De acuerdo con las realizaciones adicionales, pero puramente opcionales de la invención, el método es tal que:

- las etapas de configuración se activan automáticamente en cada relanzamiento del sistema cliente que incluye además las etapas de:

  consultar un servidor de versiones que opera en el entorno de red cliente/servidor, incluyendo la consulta una identificación del sistema cliente que incluye además un número de versión de aplicación de cliente;

obtener del servidor de versiones consultado un valor de estado para operar el sistema cliente en un modo que incluye el modo degradado y el modo válido.

- La consulta incluye además la localización geográfica, la identificación de usuario y todo tipo de información sobre el sistema cliente para establecer estadísticas.

- El contenido de la consulta se almacena en una base de datos del servidor de versiones.

- El relanzamiento del sistema cliente se fuerza por el proveedor de servicios que opera el servidor de aplicaciones cuando se pone en producción la nueva versión no compatible hacia atrás del programa de software del servidor de aplicaciones.

- El relanzamiento del sistema cliente se fuerza automáticamente cuando se carga la nueva versión no compatible hacia atrás del programa de software del servidor de aplicaciones.

- El relanzamiento del sistema cliente se fuerza automáticamente en caso de un repliegue a una versión anterior.

- La aplicación de cliente recibe del servidor de versiones el valor de estado para operar el sistema cliente en un modo no válido.

- La etapa de obtención incluye, en el servidor de versiones, la etapa previa de:

  comprobar el número de versión de la aplicación de cliente contra una meta-regla para declarar de inmediato la aplicación de cliente no válida más antigua que lo que especifica la meta-regla.

- esto incluye, si la comprobación de la meta-regla tiene éxito, las etapas adicionales de:

  comprobar el número de versión de la aplicación de cliente contra un conjunto de reglas de compatibilidad;

  declarar inválida, obsoleta o degradada la aplicación de cliente afectada por las reglas de compatibilidad de acuerdo con el contenido de la regla; de otra manera,

  declarar válida la aplicación de cliente no afectada por ninguna regla de compatibilidad.

- La etapa de obtención entrega además al sistema cliente un parche para corregir uno o más problemas de la aplicación de cliente.

- El parche se aplica dinámicamente cada vez que se relanza el sistema cliente.

- La aplicación de cliente incluye una interfaz gráfica de usuario.

La invención se refiere también a un sistema para gestionar la introducción de una nueva versión no compatible hacia atrás de un programa de software de un servidor de aplicaciones en un entorno de red cliente/servidor que incluye un servidor de versiones y una base de datos que comprende unos medios adaptados para realizar cada etapa del método.

La invención también se refiere a un producto de programa informático almacenado en un medio de almacenamiento legible por ordenador, que comprende unos medios de código legibles por ordenador para hacer que al menos un ordenador opere el método para gestionar la introducción de una nueva versión no compatible hacia atrás de un programa de software de un servidor de aplicaciones en un entorno de red cliente/servidor.

Breve descripción de los dibujos

La figura 1 describe un sistema a modo de ejemplo de acuerdo con la invención basado en un GDS que incluye un servidor de versiones y una base de datos.
La figura 2 es una vista de alto nivel de las etapas principales del método en un nodo cliente para decidir cómo debe operarse la aplicación de cliente y la GUI en función de la versión de servidor en producción.
La figura 3 trata cómo el servidor de versiones gestiona la atribución de un estado a un sistema cliente que ha emitido una consulta en el inicio de sesión.
La figura 4 describe el parcheo en línea de un sistema cliente, una mejora funcional aportada por el uso de un servidor de versiones.

Descripción detallada

La siguiente descripción detallada de la invención se refiere a los dibujos adjuntos. Aunque la descripción incluye

unas realizaciones a modo de ejemplo, son posibles otras realizaciones, y pueden hacerse cambios en las realizaciones descritas sin alejarse del espíritu y del alcance de la invención.

La figura 1 describe un sistema a modo de ejemplo de acuerdo con la invención basado en un GDS (100) como se trata en la sección de antecedentes. El GDS o cualquier sistema equivalente que proporciona servicios a través de una red en un entorno cliente/servidor que implementa servidores, en general, a partir de grandes recursos informáticos (110), para soportar una multiplicidad de aplicaciones cliente localizadas remotamente (120) (es decir, aplicaciones de software que se ejecutan a nivel de los clientes) y sistemas (122). En este ejemplo específico usado para ilustrar el sistema cliente de la invención, se usa un sistema de control de salida de aeropuerto (DCS), por ejemplo, cuando los pasajeros embarcan en los aviones. Las conexiones se mantienen (130) entre servidores y clientes a través de una red de área extensa (WAN) posiblemente comprendida de cualquier combinación de red privada y pública incluyendo Internet. Los recursos informáticos están, por ejemplo, todos interconectados con una red de área local privada o LAN (105). De manera similar, un conjunto local de aplicaciones cliente puede estar interconectado a través de una LAN, por ejemplo, una LAN de aeropuerto (125) mientras que unos conjuntos independientes de aplicaciones cliente se comunican con el GDS a través de la WAN (130). La interacción con las aplicaciones cliente se realiza en este ejemplo a través de una pasarela (101) que permite a los clientes acceder a todas las aplicaciones (109) proporcionadas por el GDS incluyendo la aplicación DCS mencionada anteriormente en el presente documento. Un sistema de este tipo incluye normalmente un servidor de inicio de sesión y de seguridad o LSS (103) dirigido a comprobar que las aplicaciones cliente que intentan conectarse son legítimas y puede proporcionar las credenciales esperadas para autorizarse a trabajar con cualquiera de las aplicaciones soportadas (109).

En la presente descripción, el programa de software del servidor de aplicaciones significa los recursos de software usados al nivel del servidor de aplicaciones para realizar sus funciones. A menos que se indique explícitamente, la expresión "programa de software del programa de aplicación del servidor de aplicaciones" se acorta a veces en el presente documento con la expresión "servidor de aplicaciones". De hecho, incluso si los servidores comprenden componentes de hardware, el objeto de la invención es gestionar las versiones de los componentes de software.

La invención presenta un servidor de versiones (105) que trabaja junto con una base de datos (107). Como se explica en detalle en la siguiente descripción, el papel del servidor de versiones es realizar un seguimiento de todas las versiones de aplicación de cliente (120) presentes en el campo. Normalmente, un GDS y los sistemas equivalentes (100) son capaces de interconectar miles de aplicaciones cliente remotas (120) a través de una WAN a nivel mundial (130). Las características del cliente se almacenan en la base de datos desde donde se recuperarán por el servidor de versiones cuando sea necesario.

Cuando la compatibilidad hacia atrás del servidor no es posible o sería demasiado costosa, el método de la invención consiste en desplegar primero a todos los nodos de clientes remotos una versión actualizada de la aplicación de cliente y la GUI. Esto implica que la nueva versión de la aplicación de cliente se hace compatible tanto con la versión actual del servidor como con la versión más reciente, es decir, la que está por venir. Con un esquema de este tipo, el nuevo servidor no necesitará ser compatible con aplicaciones cliente y las GUI más antiguas cuando se instalen.

Como consecuencia de la estrategia anterior, durante la fase de despliegue (un despliegue completo puede requerir normalmente varias semanas para completarse en una red que comprende miles de nodos cliente), la nueva aplicación de cliente, cuando está instalada, se degradada a la versión de servidor N, es decir, a la versión actual en producción. Su estado se cambia de acuerdo con la base de datos (107) del servidor de versiones (105) a ‘degradado a la versión N’. Por lo tanto, durante el período intermedio mientras la aplicación de cliente se despliega en los nodos cliente, aquellos de los nodos que ya se han actualizado están usando sistemáticamente, es decir, cada vez que se relanza el cliente, la versión N compatible con el servidor actual.

Una vez que la distribución está completa o está casi terminada, el sistema servidor, por ejemplo, el GDS (100) de la figura 1 puede decidir promocionar el nuevo servidor de tal manera que todos o al menos una mayoría de los usuarios finales puedan empezar a aprovechar las funciones y características de cliente/servidor actualizadas. La promoción puede realizarse manualmente por un administrador del sistema o puede activarse automáticamente en el momento de la carga del software. En este caso, se le indica al servidor de versiones que cambie el estado de los nodos clientes remotos a ‘válido’ en la base de datos de tal manera que la próxima versión del servidor (N + 1) pueda ponerse en producción y tener la aplicación de cliente y la GUI funcionando también al nivel N + 1. Para lograr esto, el servidor de versiones fuerza la desconexión de todas las sesiones que están usando la versión N (y de las versiones anteriores, si las hay). Para esto, el servidor de versiones envía un mensaje de cierre de sesión a cada nodo remoto cliente para finalizar las sesiones de usuario y solicitar el relanzamiento de la aplicación de cliente y la GUI. A continuación, en el próximo relanzamiento, la versión N + 1 se usa automáticamente como se trata adicionalmente en la siguiente descripción de la invención. Debería observarse en este caso, que el sistema del proveedor de servicios GDS tiene la libertad de controlar a través del servidor de versiones pocos o muchos nodos cliente según sea necesario. Solo los nodos cliente afectados por una actualización específica necesitan relanzarse. Las actualizaciones, degradaciones (por ejemplo, en el caso de un repliegue a una versión anterior) y el bloqueo de versión (si la versión actual es incompatible) pueden realizarse en diferentes niveles de granularidad. Esto puede

abarcar desde todo el mundo, una región, un país, una ciudad, un aeropuerto hasta cualquier lugar identificable, tal como un terminal específico en un aeropuerto, una puerta de embarque, una oficina, etc., proporcionando de este modo una gran cantidad de flexibilidad en el despliegue de nuevas versiones de cliente.

La figura 2 describe las etapas principales del método en un nodo cliente para decidir cómo debe operar la aplicación de cliente y la GUI en función de la versión del servidor en producción.

El proceso se ejecuta cada vez que se relanza la aplicación de cliente (210). Una transacción se inicia automáticamente (212) mediante la aplicación de cliente en el inicio de sesión para recuperar del servidor de versiones y de la base de datos (220) los detalles de la versión de aplicación de cliente que se va a usar (214). La consulta enviada automáticamente por la aplicación de cliente al servidor de versiones debe contener la identificación (ID) del nodo cliente, incluyendo el ID de aplicación durante su uso y su número de versión.

En una realización preferida de la invención, para establecer estadísticas sobre el uso de las versiones de la aplicación de cliente y el progreso del despliegue de una nueva versión, la consulta debería contener más información acerca del nodo cliente, incluyendo:

- identificación del usuario;
- oficina del usuario, por ejemplo: LONLH033 para un agente de Lufthansa (LH) localizado en Londres;
- localización del usuario, por ejemplo: LHR/T2/GTE/20 para el aeropuerto de Heathrow de Londres, terminal 2 en la puerta 20;
- organización del usuario, por ejemplo: LH, el nombre de la compañía aérea;
- etc.

Información que se proporciona de cualquier modo al servidor de inicio de sesión y de seguridad (LSS) mostrado en la figura 1 (103) como parte de las credenciales a proporcionarse de tal manera que se reconozca como un usuario legítimo del servidor de aplicaciones.

De este modo, puede usarse lo que se proporciona en la consulta (212) mediante la aplicación de cliente para obtener información invaluable sobre el despliegue de una nueva versión de aplicación en el campo y, en general, sobre las características de la población de aplicaciones cliente que interconexionan con los servidores de aplicaciones. Por ejemplo, el GDS puede establecer una lista de localizaciones en el mundo usando una cierta versión de la aplicación de cliente o puede detectar dónde hay aún demasiadas versiones antiguas en uso. La información proporcionada por la aplicación de cliente se reúne en la base de datos y puede explotarse por cualquier tipo de programa ejecutado desde un centro administrativo encargado de observar y gestionar una gran población de nodos clientes remotos.

El servidor de versiones (220) mantiene la gestión de una lista de estados de la aplicación de cliente y las GUI en uso en todos los nodos cliente. A continuación, conociendo la versión del servidor de aplicaciones actualmente en producción y, como se ha explicado anteriormente, sobre la base de la información proporcionada por los nodos cliente en las consultas, se devuelve un estado (214) a cada sistema cliente en el momento del inicio de sesión. El estado puede tomar uno de los siguientes valores:

Válido:	es el estado de la última versión descargada (N + 1) de la aplicación de cliente y la GUI. Este es el estado normal después de que se haya desplegado una nueva aplicación de cliente y se haya puesto en producción la nueva versión correspondiente del servidor.
Obsoleto:	dice que la aplicación de cliente y la GUI es una versión anterior pero aún compatible con el servidor de producción actual. Este estado se usa cuando una nueva versión de la GUI está disponible pero la GUI desplegada más antigua permanece sin embargo compatible con el servidor actual. En función de la configuración del cliente, se muestra un mensaje de advertencia opcionalmente al usuario final en el inicio de sesión.
Degradado a la versión N:	indica que la aplicación de cliente recientemente desplegada y la versión de la GUI N + 1 deben comportarse como la versión N (mientras se está distribuyendo una nueva aplicación de cliente y de la GUI, antes de activar el nuevo servidor correspondiente). Como se ha tratado anteriormente, este estado se usa principalmente para permitir gestionar la activación de servidores no compatibles hacia atrás. También se usa ocasionalmente para gestionar un repliegue de un servidor (a una versión anterior) como se explica a continuación.
Inválido:	dice que la aplicación de cliente y la GUI es una versión antigua NO compatible con el servidor de producción actual. El usuario final del nodo cliente recibe una nota de advertencia al iniciar sesión y normalmente se bloquea.

Los valores de estado anteriores se interpretan por los sistemas de cliente (216) con el fin de comportarse de acuerdo con la versión del servidor de aplicaciones en producción. Después de lo cual pueden realizarse transacciones normales (218) entre las aplicaciones cliente y el servidor de aplicaciones (230) encargado de manejarlas.

Si, por cualquier razón, un servidor de aplicaciones que se ha promovido encuentra un problema y debe eliminarse, debe producirse un repliegue a una versión anterior. A continuación, la nueva aplicación de cliente y la GUI que ya se han descargado deben establecerse de nuevo al estado "degradado" de manera que se vuelvan a comportar, como en el período intermedio mientras el despliegue estaba en progreso, de acuerdo con la versión de servidor anterior. Si algunos sistemas de cliente seguían usando la versión anterior de la aplicación de cliente y de la GUI, su estado se invierte desde 'obsoleto' a 'válido'.

La figura 3 trata cómo el servidor de versiones gestiona atribuir un estado a un sistema cliente que ha emitido una consulta en el inicio de sesión.

El proceso comienza con la versión (310) de la aplicación de cliente y la GUI proporcionada en la consulta emitida al servidor de versiones por el sistema cliente. Para decidir qué estado debe devolverse cuando se interroga por un servidor de versiones de sistema cliente, se usan unas reglas de compatibilidad comprobándose cada una (340) contra la versión de cliente proporcionada (310), sabiendo qué versión del servidor de aplicaciones está en producción. Sin embargo, para evitar tener una proliferación de reglas de compatibilidad hay una comprobación previa de la versión del cliente contra una meta-regla (320). La meta-regla se usa para eliminar directamente todos los números de versión que son más antiguos que un valor dado. Hay una meta-regla por aplicación de cliente. Por lo tanto, si la comprobación de la meta-regla falla (331), se devuelve (352) directamente el 'estado inválido'. De lo contrario, si la versión del cliente pasa la comprobación (332) de la meta-regla, es necesario comprobar (340) las reglas de compatibilidad. Las reglas de compatibilidad realizan un seguimiento de todas las situaciones donde existe incompatibilidad entre el servidor en producción y los sistemas de cliente específicos de tal manera que todos los no afectados por las reglas se declaren finalmente válidos (380). De lo contrario, los sistemas de cliente afectados por las reglas se declaran como 'inválidos' (350), 'obsoletos' (360) o 'degradados' (370) de acuerdo con el contenido de la regla que se les aplica.

La figura 4 describe el parcheo en línea de un sistema cliente, una mejora funcional aportada por el uso de un servidor de versiones.

El uso de un servidor de versiones es específicamente útil en grandes redes que implican numerosos sistemas de cliente que afectan posiblemente a miles o decenas de miles de nodos clientes. Entonces, el despliegue de una nueva aplicación de cliente es una tarea pesada y larga que normalmente tarda semanas en completarse. Debido a que los sitios remotos pueden no estar bajo el control directo del proveedor de servicios, por ejemplo, los GDS usados para ilustrar la invención, los responsables, por ejemplo, las autoridades aeroportuarias, podrían estar reacios a instalar una nueva versión de la aplicación por miedo a impactar negativamente su sistema. El mecanismo de repliegue soportado por la invención es una respuesta a esta preocupación permitiendo, en caso de un problema grave volver a una versión anterior del servidor. Un servidor de versiones de acuerdo con la invención permite implementar una mejora funcional adicional para eludir un problema que se encontraría mientras el despliegue de la aplicación de cliente está en curso o en cualquier momento después de que se haya activado.

Si se descubre un problema importante de bloqueo en una versión de aplicación de cliente (410) que ya se ha desplegado a nivel mundial, el servidor de versiones (420) puede proporcionar un pedazo de código, un parche, para resolver el problema sin tener que redistribuir toda la aplicación de cliente. Para hacer esto, tal como ya se ha explicado en la figura 2, cuando el servidor de versiones responde a la consulta (412) enviada automáticamente al inicio de sesión de un sistema cliente remoto, se agrega un parche en la respuesta. Sobre la base del número de versión y el nombre de aplicación proporcionados en la consulta, el servidor de versiones reconoce que la versión de aplicación de cliente tiene un problema funcional. A continuación, la respuesta reenviada por el servidor de versiones incluye un estado, ya tratado, y un parche del problema funcional (414). A la recepción de la respuesta del servidor, la aplicación de cliente cambia su comportamiento de acuerdo con el estado de la versión y aplica el parche sobre sí misma (416). El parche no se almacena permanentemente ni se instala en la máquina de aplicación de cliente, sino que se aplica cada vez que se reinicia la aplicación de cliente. Después de lo cual las transacciones normales (418) pueden reanudarse entre las aplicaciones cliente y el servidor de aplicaciones (430) encargado de manejarlas. Este modo de funcionamiento continúa hasta que se redistribuye una nueva versión.

REIVINDICACIONES

1. Un método para gestionar la introducción de una nueva versión no compatible hacia atrás de un programa de software que se ejecuta en un servidor de aplicaciones (109) en un entorno de red cliente/servidor (130), comprendiendo el método:

en primer lugar, distribuir a una pluralidad de sistemas de cliente (122) a servir por la nueva versión no compatible hacia atrás del programa de software del servidor de aplicaciones, una aplicación de cliente (120) operable en un modo degradado compatible con la versión actual del programa de software del servidor de aplicaciones y en un modo válido compatible con la nueva versión no compatible hacia atrás de dicho programa de software;
configurar, tras la instalación en los sistemas de cliente, la aplicación de cliente en un modo degradado (370) compatible con la versión actual del programa de software del servidor de aplicaciones;
mantener operativa la aplicación de cliente de cada sistema cliente en el modo degradado hasta la introducción de la nueva versión no compatible hacia atrás del programa de software del servidor de aplicaciones;
caracterizado por que el método comprende, además:

configurar, tras la introducción de la nueva versión no compatible hacia atrás del programa de software del servidor de aplicaciones, la aplicación de cliente de cada sistema cliente en el modo válido (380) compatible con la nueva versión no compatible hacia atrás del programa de software del servidor de aplicaciones, incluyendo la configuración, dando instrucciones a un servidor de versiones (105), operando el servidor de versiones (105) en el entorno de red cliente/servidor y trabajando junto con una base de datos (107), para cambiar en la base de datos (107) un valor de estado del sistema cliente (216) que corresponde al modo válido;
consultar (212) cada vez que se relanza la aplicación de cliente (120) el servidor de versiones (105), incluyendo la consulta una identificación del sistema cliente (216) que incluye además un número de versión de aplicación de cliente; y
obtener (214) del servidor de versiones consultado (105) el valor de estado correspondiente al modo válido y operar, a partir de ese momento, la aplicación de cliente (120) del sistema cliente (216) en el modo válido.

2. El método de la reivindicación 1, en el que la consulta (212) incluye además la localización geográfica, la identificación del usuario y la información sobre el sistema cliente para establecer estadísticas.

3. El método de la reivindicación 1 o 2, en el que el contenido de la consulta se almacena en una base de datos (107) del servidor de versiones.

4. El método de la reivindicación 1, en el que se fuerza el relanzamiento del sistema cliente por un proveedor de servicios que opera el servidor de aplicaciones (100) cuando se pone en producción la nueva versión no compatible hacia atrás del programa de software del servidor de aplicaciones.

5. El método de la reivindicación 4, en el que se fuerza automáticamente el relanzamiento del sistema cliente cuando se carga la nueva versión no compatible hacia atrás del programa de software del servidor de aplicaciones.

6. El método de la reivindicación anterior, en el que se fuerza el relanzamiento enviando un mensaje de cierre de sesión y una solicitud de relanzamiento desde el servidor de versiones al sistema cliente.

7. El método de la reivindicación 4, en el que se fuerza automáticamente el relanzamiento del sistema cliente en el caso de un repliegue a una versión anterior del programa de software del servidor de aplicaciones.

8. El método de la reivindicación 1, en el que la aplicación de cliente recibe desde el servidor de versiones el valor de estado para operar el sistema cliente en un modo no válido (350).

9. El método de la reivindicación 1, en el que la etapa de obtención incluye, en un servidor de versiones, la etapa previa de:

comprobar el número de versión de la aplicación de cliente contra una meta-regla (320) para declarar de inmediato la aplicación de cliente no válida más antigua que lo que especifica la meta-regla.

10. El método de la reivindicación 9, que incluye, si la comprobación de la meta-regla tiene éxito, las etapas adicionales de:

comprobar el número de versión de la aplicación de cliente contra un conjunto de reglas de compatibilidad (340);
declarar inválida (350), obsoleta (360) o degradada (370) la aplicación de cliente afectada por las reglas de compatibilidad (340) de acuerdo con el contenido de la regla; de otra manera,
declarar válida (380) la aplicación de cliente no afectada por ninguna regla de compatibilidad.

11. El método de la reivindicación 1, en el que la etapa de obtención entrega adicionalmente al sistema cliente un parche (414) para corregir uno o más problemas de la aplicación de cliente.

12. El método de la reivindicación 11, en el que el parche se aplica de manera dinámica cada vez que se relanza (416) el sistema cliente.

13. El método de una cualquiera de las reivindicaciones anteriores, en el que la aplicación de cliente incluye una interfaz gráfica de usuario (210).

14. Un sistema para gestionar la introducción de una nueva versión no compatible hacia atrás de un programa de software de un servidor de aplicaciones en un entorno de red cliente/servidor (100) que incluye un servidor de versiones (105) y una base de datos (107) que comprende unos medios adaptados para realizar cada etapa del método de acuerdo con una cualquiera de las reivindicaciones anteriores.

15. Un producto de programa informático almacenado en un medio de almacenamiento legible por ordenador, que comprende unos medios de código legibles por ordenador para hacer que al menos un ordenador (110, 122) opere el método para gestionar la introducción de una nueva versión no compatible hacia atrás de un programa de software de un servidor de aplicaciones en un entorno de red cliente/servidor de acuerdo con una cualquiera de las reivindicaciones 1 a 13.

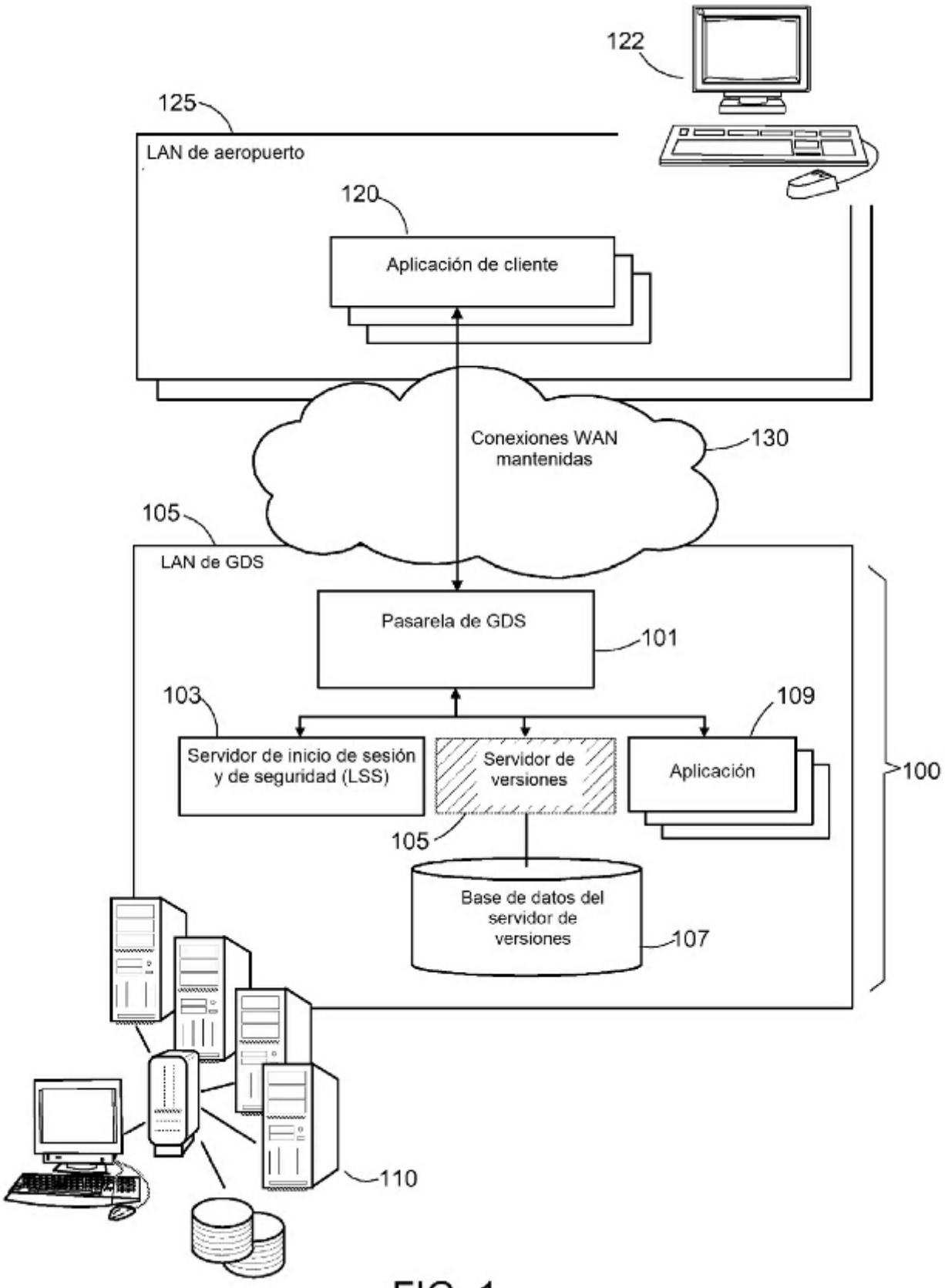
122
125
LAN de aeropuerto
120
Aplicación de cliente
Conexiones WAN mantenidas
130
105
LAN de GDS
Pasarela de GDS
101
103
109
Servidor de inicio de sesión y de seguridad (LSS)
Servidor de versiones
Aplicación
100
105
Base de datos del servidor de versiones
107
110

FIG. 1

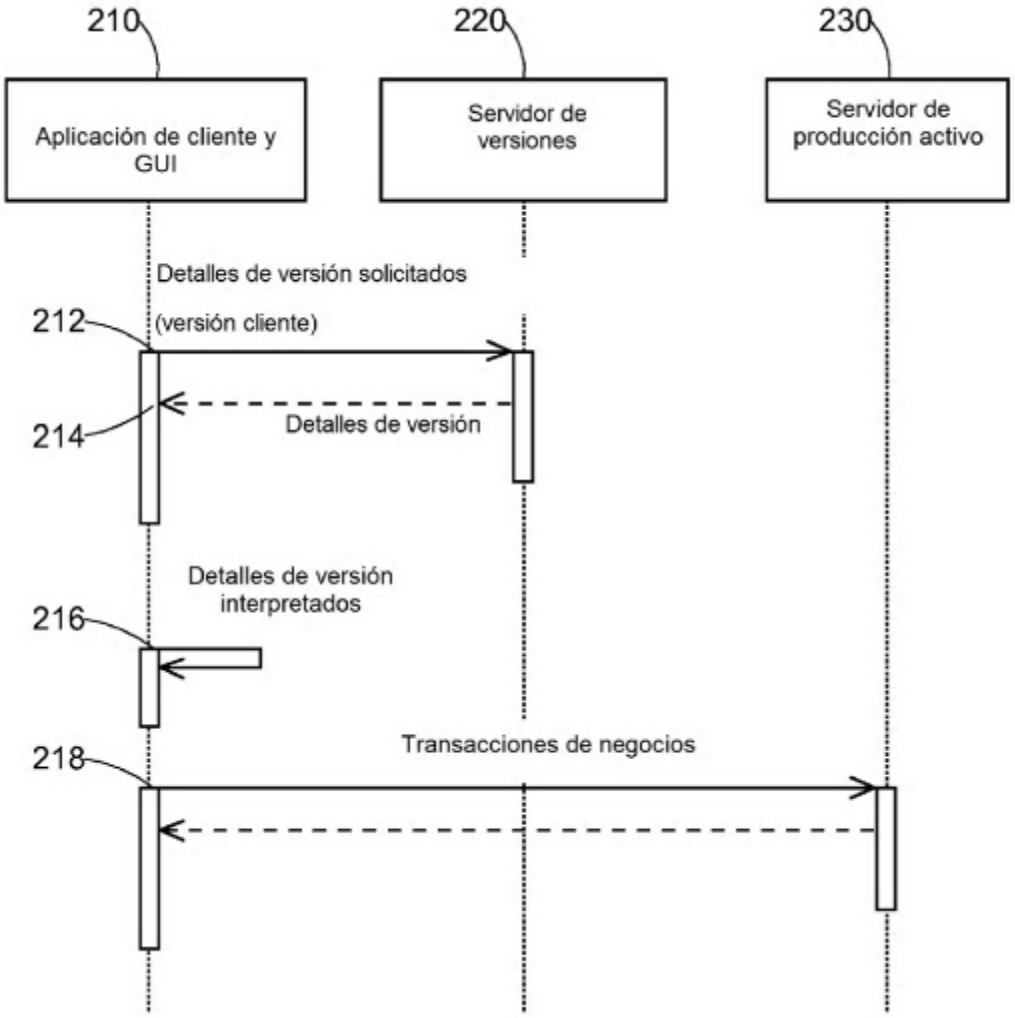
210
220
230
Aplicación de cliente y GUI
Servidor de versiones
Servidor de producción activo
Detalles de versión solicitados
(versión cliente)
212
214
Detalles de versión
Detalles de versión interpretados
216
Transacciones de negocios
218

FIG. 2

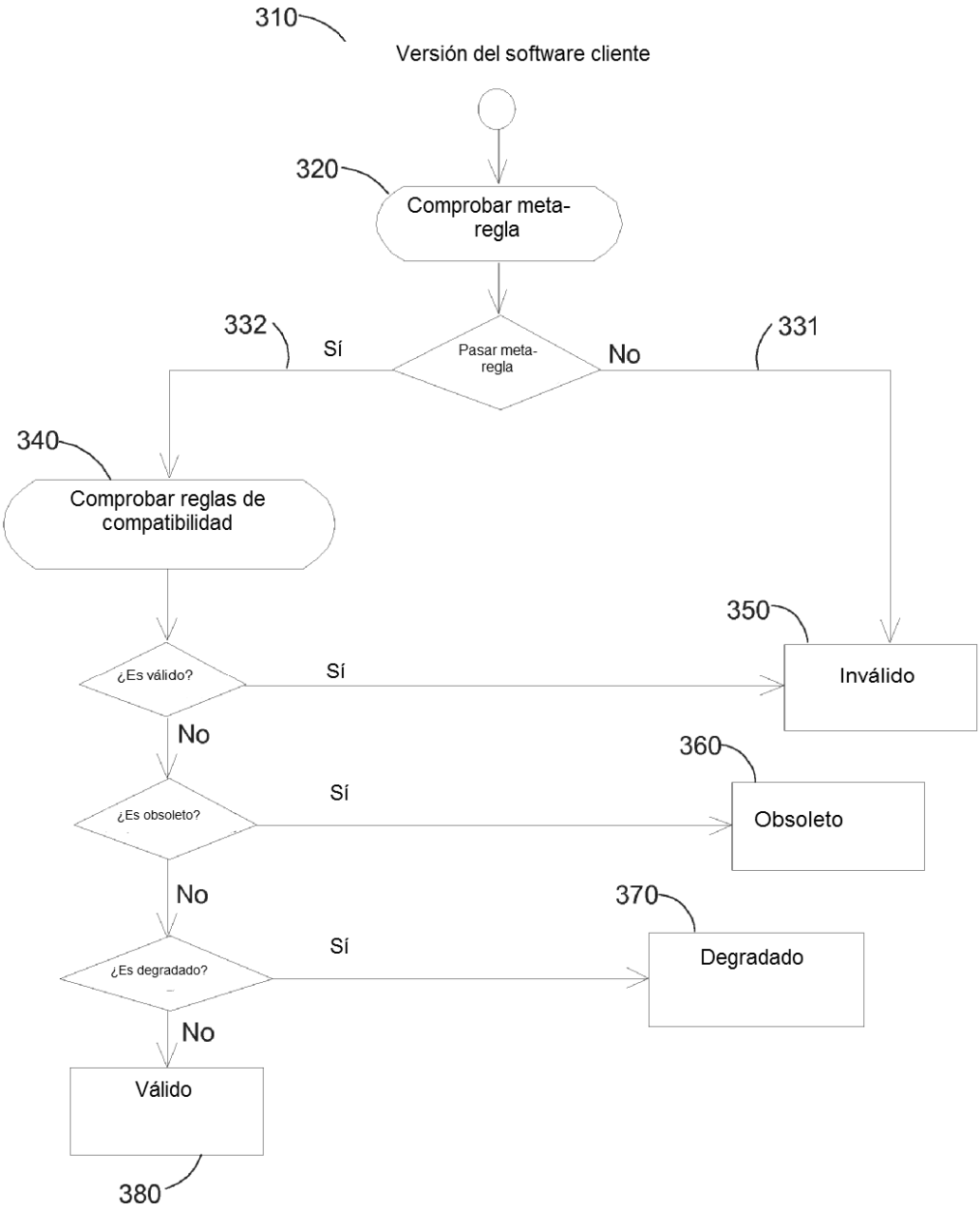
310
Versión del software cliente
320
Comprobar meta-regla
332
Sí
Pasar meta-regla
No
331
340
Comprobar reglas de compatibilidad
350
¿Es válido?
Sí
Inválido
No
360
¿Es obsoleto?
Sí
Obsoleto
No
370
¿Es degradado?
Sí
Degradado
No
Válido
380

Figura 3

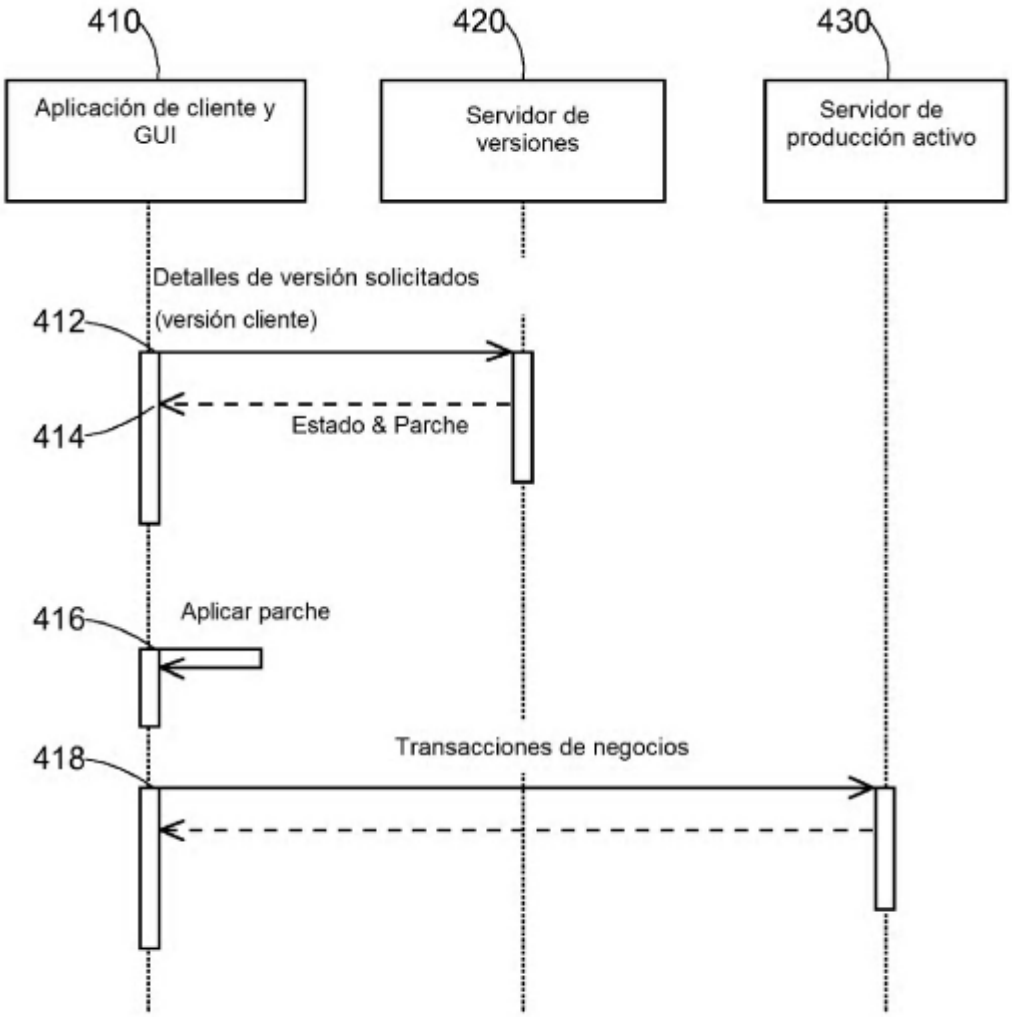
410
420
430
Aplicación de cliente y GUI
Servidor de versiones
Servidor de producción activo
Detalles de versión solicitados
(versión cliente)
412
414
Estado & Parche
416
Aplicar parche
418
Transacciones de negocios

FIG. 4