(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 673 199**

(51) Int. CI.:

G06F 21/52 (2013.01)
G06F 21/62 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **14.10.2005 PCT/IB2005/003077**

(87) Fecha y número de publicación internacional: **27.04.2006 WO06043143**

(96) Fecha de presentación y número de la solicitud europea: **14.10.2005 E 05818855 (8)**

(97) Fecha y número de publicación de la concesión europea: **11.04.2018 EP 1817668**

(54) Título: **Terminal, método y producto de programa informático para validar una aplicación de software**

(30) Prioridad:

20.10.2004 US 969145

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
20.06.2018

(73) Titular/es:

NOKIA TECHNOLOGIES OY (100.0%)
Karaportti 3
02610 Espoo, FI

(72) Inventor/es:

KOTAMARTHI, PADMAJABALA y
NARAYANAN, RAM GOPAL LAKSHMI

(74) Agente/Representante:

VALLEJO LÓPEZ, Juan Pedro

DESCRIPCIÓN

Terminal, método y producto de programa informático para validar una aplicación de software

Campo de la invención

La presente invención se refiere en general a terminales y métodos de validación de una aplicación de software y, más particularmente, se refiere a terminales, métodos, y productos de programa informático para validar aplicaciones de software en un entorno de sistema operativo de terminal móvil.

Antecedentes de la invención

Como es bien conocido, puede usarse diverso equipo de usuario, incluyendo ordenadores (fijos o portátiles), teléfonos móviles, asistentes de datos personales (PDA), organizadores y similares, para comunicar con otro equipo de usuario en un sistema de comunicación o para acceder a la Internet para obtener servicios. El equipo de usuario móvil a menudo se denomina como un terminal móvil, y puede definirse como un medio que puede realizar comunicación mediante una interfaz inalámbrica con otro dispositivo tal como una estación base de una red de telecomunicación móvil o cualquier otra estación, incluyendo otros terminales móviles. Tales terminales pueden adaptarse para comunicación de voz, mensaje de texto o datos mediante la interfaz inalámbrica.

A medida que la convergencia digital cambia la manera en la que las empresas y usuarios usan y comparten información, está teniendo lugar una unión de tecnología de comunicación digital, medios digitales e informática. Y el equipo de usuario se está desarrollando para posibilitar que los usuarios accedan a las facilidades proporcionadas por tal convergencia. En este sentido, los denominados teléfonos móviles de gama alta (o teléfonos inteligentes o comunicadores) son ejemplos de terminales móviles que se han desarrollado para cumplir este requisito. Estos teléfonos móviles no únicamente realizan funciones de un teléfono móvil convencional, sino también pueden realizar funciones previamente limitadas a ordenadores personales, asistentes digitales personales o similares. Por ejemplo, un número de estos teléfonos móviles incluyen funcionalidad para instalar o cargar software de terceros en los teléfonos móviles.

Como también es bien conocido por los expertos en la materia, diversas piezas de software pueden ser maliciosas en apariencia de virus, caballos de Troya o algún otro elemento introducido por un pirata para abusar del teléfono móvil u obtener acceso de otra manera a funcionalidad del teléfono móvil. Es bien conocido que los ordenadores personales (PC) son susceptibles de tales problemas, y se han propuesto soluciones en el campo de la informática personal para superarlos, incluyendo implementar medidas de seguridad detalladas para evitar, detectar y/o responder a los riesgos de seguridad planteados por tal software malicioso. Sin embargo, los sistemas operativos abiertos de la mayoría de los dispositivos móviles, tal como el sistema operativo (SO) actual Symbian™, no proporcionan un modelo de seguridad con buena resolución para superar el riesgo de seguridad. Adicionalmente diversas soluciones que se han propuesto son soluciones de nivel de aplicación que están basadas en software, y como tal, pueden omitirse por un programador sofisticado.

El documento EP 1465041 se refiere a un dispositivo de comunicación, método y programa para comprobar seguridad de permiso de ejecución de software para aplicaciones de MIDP que se analiza en el documento 'Mobile Information Device Profile for Java™ 2 MicroEdition'; JSR 118 Expert Group; Java Community Press; 5 de noviembre de 2002.

Sumario de la invención

De acuerdo con aspectos de la presente invención se proporciona un método de acuerdo con la reivindicación 1, un medio de almacenamiento legible por ordenador de acuerdo con la reivindicación 8, y un aparato de acuerdo con la reivindicación 9.

A la luz de los antecedentes anteriores, las realizaciones de la presente invención proporcionan un terminal, método y producto de programa informático mejorados para validar una aplicación de software. De acuerdo con realizaciones de la presente invención, se proporciona una estructura de seguridad mediante la cual puede comprobarse la autenticidad, integridad y/o autoridad de las aplicaciones recibidas por el terminal antes de instalar, cargar y/o proporcionar servicios a estas aplicaciones. Más particularmente, por ejemplo, las aplicaciones instaladas o cargadas de otra manera en un terminal están asociadas a registros de permiso que identifican servicios que se permite que las aplicaciones reciban desde una plataforma de SO que opera por debajo de las aplicaciones, incluyendo, por ejemplo, las operaciones que se permite que se realicen en la plataforma de SO. A continuación, cuando una aplicación solicita un servicio proporcionado por la plataforma de SO, puede verificarse la autoridad de la aplicación para que reciba el servicio basándose en el registro de permiso asociado a la respectiva aplicación antes de que la aplicación reciba el servicio solicitado. Por lo tanto, puede evitarse que las aplicaciones no autorizadas, tales como aplicaciones maliciosas, accedan a servicios para los que las respectivas aplicaciones no tienen autoridad expresa para acceder, como se identifica en los registros de permisos.

De acuerdo con un aspecto de la presente invención, se proporciona un terminal para validar una aplicación de software. El terminal incluye un procesador que puede operar una plataforma (por ejemplo, plataforma de SO Symbian™) de sistema operativo (SO), y que puede operar al menos una aplicación de software por encima de la plataforma de SO. La aplicación o aplicaciones de software están asociadas a un registro de permiso que incluye permisos que identifican servicios a los que la aplicación de software está autorizada a recibir desde la plataforma de SO. La plataforma de SO puede recibir una solicitud, desde una aplicación de software, para un servicio de la plataforma de SO. La plataforma de SO, o un módulo de verificación de la plataforma de SO, puede verificar la autoridad de la aplicación de software determinando si la aplicación de software está autorizada a recibir el servicio solicitado basándose en el registro de permiso asociado. Y si la aplicación de software está autorizada, la plataforma de SO puede proporcionar el servicio solicitado a la aplicación de software.

El terminal puede recibir una pluralidad de solicitudes desde la aplicación de software para al menos un servicio de la plataforma de SO. Para cada servicio solicitado, a continuación, la plataforma de SO puede determinar si la aplicación de software está autorizada a recibir el servicio solicitado basándose en el registro de permiso asociado. La plataforma de SO puede proporcionar a continuación, a la aplicación de software, aquellos servicios para los que la aplicación de software está autorizada.

Además de verificar la autoridad de la aplicación de software, la plataforma de SO puede verificar adicionalmente una autenticidad de la aplicación de software. Más particularmente, la plataforma de SO o una aplicación de instalador de la plataforma de SO puede verificar la autenticidad de la aplicación de software basándose en un origen de la aplicación de software, donde la aplicación de software se ha recibido anteriormente desde el origen. Independientemente de cómo la plataforma de SO verifica la autenticidad de la aplicación de software, sin embargo, la plataforma de SO puede posteriormente instalar la aplicación de software para operación en el terminal si se verifica la autenticidad de la aplicación de software. En tales casos, la plataforma de SO puede verificar la autenticidad e instalar la aplicación de software antes de recibir una solicitud desde la aplicación de software para un servicio de la plataforma de SO. Después de verificar la autenticidad de la aplicación, sin embargo, la plataforma de SO puede crear adicionalmente el registro de permiso para la aplicación de software si se verifica la autenticidad de la aplicación de software, y posteriormente almacenar el registro de permiso en una base de datos de políticas de la plataforma de SO.

Además, la plataforma de SO puede adicionalmente iniciar la carga de la aplicación de software, y verificar una integridad de la aplicación de software antes de cargar la aplicación de software. En este sentido, el registro de permiso asociado a la aplicación de software puede incluir adicionalmente una firma asociada a la aplicación de software. En tales casos, la plataforma de SO puede verificar la integridad de la aplicación de software generando una firma de verificación basándose en la aplicación de software, comparar la firma de verificación con la firma en el registro de permiso asociado a la aplicación de software, y posteriormente verificar la integridad de la aplicación de software basándose en la comparación. Independientemente de cómo se verifica la integridad de la aplicación de software, sin embargo, la plataforma de SO o una aplicación de cargador de la plataforma de SO puede cargar la aplicación de software para operación en el terminal si se verifica la integridad de la aplicación de software. A continuación, después de cargar la aplicación de software, la plataforma de SO puede recibir una solicitud desde la aplicación de software para un servicio de la plataforma de SO.

De acuerdo con otros aspectos de la presente invención, se proporciona un método y producto de programa informático para validar una aplicación de software. Las realizaciones de la presente invención por lo tanto proporcionan un terminal, método y producto de programa informático mejorados para validar una aplicación de software. Como se ha indicado anteriormente, y se explica a continuación, las realizaciones de la presente invención proporcionan una estructura de seguridad mediante la cual se verifica la autoridad de una aplicación de software para recibir un servicio antes de proporcionar ese servicio a la aplicación de software. También de acuerdo con realizaciones de la presente invención, la autenticidad de la aplicación de software puede verificarse antes de instalar la aplicación de software para operación en el terminal. Además, la integridad de la aplicación de software puede verificarse antes de cargar la aplicación de software para operación en el terminal, de acuerdo con realizaciones de la presente invención. Como tal, el terminal, método y producto de programa informático de las realizaciones de la presente invención resuelven los problemas identificados por técnicas anteriores y proporcionan ventajas adicionales.

Breve descripción de los dibujos

Habiendo descrito por lo tanto la invención en términos generales, se hará ahora referencia a los dibujos adjuntos, que no están dibujados necesariamente a escala, y en los que:

La Figura 1 es un diagrama de bloques de un tipo de terminal y sistema que se beneficiarían de las realizaciones de la presente invención;

La Figura 2 es un diagrama de bloques esquemático de una plataforma de sistema operativo entre una plataforma de capa de aplicación y un hardware específico de fabricante en una capa inferior;

La Figura 3 es un diagrama de bloques esquemático que ilustra funcionalmente dos aplicaciones que operan por encima de una plataforma de sistema operativo de acuerdo con una técnica convencional;

La Figura 4 es un diagrama de bloques esquemático que ilustra funcionalmente dos aplicaciones que operan por encima de una plataforma de sistema operativo de acuerdo con una realización de la presente invención; y
Las Figuras 5, 6a y 6b son diagramas de flujo que ilustran diversas etapas en un método de validación de una aplicación de software, de acuerdo con una realización de la presente invención.

Descripción detallada de la invención

La presente invención se describirá ahora más completamente en lo sucesivo con referencia a los dibujos adjuntos, en los que se muestran realizaciones preferidas de la invención. Esta invención puede realizarse, sin embargo, en muchas formas diferentes y no debería interpretarse como que está limitada a las realizaciones expuestas en el presente documento; en su lugar, estas realizaciones se proporcionan de modo que esta divulgación será minuciosa y completa, y transmitirá completamente el alcance de la invención a los expertos en la materia. Números de referencia similares se refieren a elementos similares a lo largo de todo el presente documento.

Haciendo referencia a la Figura 1, se proporciona una ilustración de un tipo de sistema y terminal que se beneficiarían de la presente invención. El sistema, método y producto de programa informático de las realizaciones de la presente invención se describirán principalmente en conjunto con aplicaciones de comunicaciones móviles. Debería entenderse, sin embargo, que el sistema, método y producto de programa informático de las realizaciones de la presente invención puede utilizarse en conjunto con diversas otras aplicaciones, tanto en la industria de las comunicaciones móviles como fuera de la industria de las comunicaciones móviles. Por ejemplo, el sistema, método y producto de programa informático de las realizaciones de la presente invención pueden utilizarse en conjunto con aplicaciones de red alámbrica y/o inalámbrica (por ejemplo, Internet).

También, debería entenderse que aunque el terminal puede ilustrarse y en lo sucesivo describirse como que comprende un teléfono móvil, los teléfonos móviles son meramente ilustrativos de un tipo de terminal que se beneficiaría de la presente invención y, por lo tanto, no deberían tomarse para limitar el alcance de la presente invención. Aunque se ilustran varias realizaciones del terminal y se describirán en lo sucesivo para fines de ejemplo, otros tipos de terminales, tales como asistentes digitales personales (PDA), buscapersonas, ordenadores portátiles y otros tipos de sistemas electrónicos, pueden emplear fácilmente la presente invención.

Como se muestra, el sistema puede incluir un número de los mismos o diferentes terminales 10 (se muestra uno). Cada terminal puede tener una antena 12 para transmitir señales a y para recibir señales desde un sitio base o estación base (BS) 14. La estación base es una parte de una o más redes celulares o móviles que cada una incluyen elementos requeridos para operar la red, tal como un centro de conmutación móvil (MSC) 16. Como es bien conocido para los expertos en la materia, la red móvil puede denominarse también como una estación base/MSC/función de interfuncionamiento (BMI). En la operación, el MSC puede encaminar llamadas, datos o similares a y desde los terminales cuando estos terminales están realizando y recibiendo llamadas, datos o similares. El MSC puede proporcionar también una conexión a partes troncales de línea terrestre cuando los terminales se ven implicados en una llamada. Además, el MSC puede controlar el reenvío de mensajes a y desde terminales, y puede controlar también el reenvío de mensajes para tales terminales a y desde un centro de mensajería (MC) 18, tal como mensajes del Servicio de Mensajes Cortos (SMS) a y desde un centro de SMS (SMSC) y/o mensajes de Servicio de Mensajería Multimedia (MMS) a y desde un centro de MMS (MMSC).

El MSC 16 puede acoplarse a una red de datos, tal como una red de área local (LAN), una red de área metropolitana (MAN), y/o una red de área extensa (WAN). El MSC puede acoplarse directamente a la red de datos. En una realización típica, sin embargo, el MSC está acoplado a una pasarela (GTW) 20, y la GTW está acoplada a una WAN, tal como internet 22. A su vez, los dispositivos tales como elementos de procesamiento (por ejemplo, ordenadores personales, ordenadores de servidor o similares) pueden acoplarse a los terminales 10 mediante la Internet (y/o directamente mediante otros medios para compartir y/u obtener datos, como se explica a continuación). Por ejemplo, como se explica a continuación, los elementos de procesamiento pueden incluir uno o más elementos de procesamiento asociados a uno o más servidores de origen 24, uno de los cuales se ilustra en la Figura 1. Los elementos de procesamiento pueden comprender cualquiera de un número de dispositivos de procesamiento, sistemas o similares que pueden operar de acuerdo con realizaciones de la presente invención. En este sentido, los elementos de procesamiento pueden comprender, por ejemplo, sistemas informáticos de servidor, sistemas informáticos de sobremesa, sistemas informáticos de portátil o similares.

Aunque no se muestra y describe cada elemento de cada posible red en el presente documento, debería apreciarse que los terminales 10 pueden acoplarse a uno o más de cualquiera de un número de diferentes redes. En este sentido, la red o redes móviles pueden soportar comunicación de acuerdo con uno cualquiera o más de un número de protocolos de comunicación móviles de la primera generación (1G), de la segunda generación (2G), 2,5G y/o de la tercera generación (3G) o similares. Adicionalmente o como alternativa, la red o redes móviles pueden soportar comunicación de acuerdo con uno cualquiera o más de un número de diferentes redes de difusión digitales, tales como redes de difusión de vídeo digital (DVB) incluyendo DVB-T (DVB-terrestre) y/o DVB-H (DVB-portátil), redes de difusión digital de servicios integrados (ISDB) incluyendo ISDB-T (ISDB-terrestre), o similares.

Más particularmente, por ejemplo, uno o más terminales 10 pueden acoplarse a una o más redes que pueden

soportar comunicación de acuerdo con protocolos de comunicación inalámbricos de la 2G IS-136 (TDMA), GSM e IS-95 (CDMA). También, por ejemplo, una o más de la red o redes pueden soportar comunicación de acuerdo con protocolos de comunicación inalámbricos de la 2,5G, GPRS, Entorno de GSM de Datos Mejorado (EDGE), o similares. Además, por ejemplo, una o más de la red o redes pueden soportar comunicación de acuerdo con protocolos de comunicación inalámbricos de la 3G tal como la red del Sistema de Telefonía Móvil Universal (UMTS) que emplea tecnología de acceso de radio de Acceso Múltiple por División de Código de Banda Ancha (WCDMA). Alguna red o redes de banda estrecha AMPS (NAMPS), así como TACS, pueden beneficiarse también de las realizaciones de la presente invención, como lo harían los terminales de modo dual o superior (por ejemplo, teléfonos digitales/analógicos o TDMA/CDMA/analógicos).

Como se muestra, además de una antena **12,** el terminal móvil **10** puede incluir un transmisor **26,** receptor **28,** y controlador **30** u otro procesador que proporciona señales a y recibe señales desde el transmisor y receptor, respectivamente. Estas señales incluyen información de señalización de acuerdo con la norma de interfaz aérea del sistema celular aplicable, y también datos de voz del usuario y/o generados por el usuario. En este sentido, el terminal puede operar con una o más normas de interfaz aérea, protocolos de comunicación, tipos de modulación y tipos de acceso. Más particularmente, el terminal puede operar de acuerdo con cualquiera de un número de protocolos de comunicación 1G, 2G, 2,5G y/o 3G o similares, tal como uno cualquiera o más de aquellos anteriormente indicados.

Se entiende que el controlador **30** incluye la circuitería requerida para implementar las funciones de audio y lógicas del terminal **10.** Por ejemplo, el controlador puede estar comprendido de un dispositivo de procesador de señales digitales, un dispositivo de microprocesador, y diversos convertidores de analógico a digital, convertidores de digital a analógico, y otros circuitos de soporte. Las funciones de control y procesamiento de señal del terminal están asignadas entre estos dispositivos de acuerdo con sus respectivas capacidades. El controlador puede incluir adicionalmente un codificador de voz interno (VC) **30a,** y puede incluir un módem de datos interno (DM) **30b.** Además, el controlador puede incluir la funcionalidad para operar uno o más programas cliente de software tal como aquellos anteriormente indicados, que pueden almacenarse en memoria (descritos a continuación).

El terminal **10** también comprende una interfaz de usuario que incluye un auricular o altavoz convencional **32,** un timbre **34,** un micrófono **36,** una pantalla **38,** y una interfaz de entrada de usuario, todos los cuales están acoplados al controlador **30.** Aunque no se muestra, el terminal puede incluir una batería para alimentar los diversos circuitos que se requieren para operar el terminal, así como proporcionar opcionalmente vibración mecánica como una salida detectable. La interfaz de entrada de usuario, que también permite que el terminal reciba datos, puede comprender cualquiera de un número de dispositivos que permiten que el terminal reciba datos, tal como un teclado numérico **40,** una pantalla táctil (no mostrada) u otro dispositivo de entrada. En las realizaciones que incluyen un teclado numérico, el teclado numérico incluye las teclas numéricas convencionales (0-9) y las relacionadas (#, *), y otras usadas para operar el terminal.

El terminal **10** puede incluir también uno o más medios para compartir y/u obtener datos. Por ejemplo, el terminal puede incluir un transceptor o interrogador de frecuencia de radio (RF) de corto alcance **42** de modo que pueden compartirse datos con y/u obtenerse desde dispositivos electrónicos de acuerdo con técnicas de RF. El terminal puede incluir adicionalmente, o como alternativa, otros transceptores de corto alcance, tal como, por ejemplo un transceptor de infrarrojos (IR) **44,** y/o un transceptor Bluetooth (BT) 46 que opera usando tecnología inalámbrica de la gama Bluetooth desarrollada por el Grupo de Interés Especial de Bluetooth. El terminal puede transmitir, adicionalmente o como alternativa, datos a y/o recibir datos desde dispositivos electrónicos de acuerdo con tales técnicas. Aunque no se muestra, el terminal puede, adicionalmente o como alternativa, transmitir y/o recibir datos desde dispositivos electrónicos de acuerdo con un número de diferentes técnicas de interconexión en red inalámbrica, incluyendo técnicas WLAN tales como técnicas IEEE 802.11 o similares.

El terminal **10** puede incluir adicionalmente memoria, tal como un módulo de identidad de abonado (SIM) **48,** un módulo de identidad de usuario extraíble (R-UIM) o similares, que normalmente almacena elementos de información relacionados con un abonado móvil. Además del SIM, el terminal puede incluir otra memoria extraíble y/o fija. En este sentido, el terminal puede incluir memoria volátil **50,** tal como Memoria de Acceso Aleatorio (RAM) volátil que incluye un área de caché para el almacenamiento temporal de datos. El terminal puede incluir también otra memoria no volátil **52,** que puede embeberse y/o puede ser extraíble. La memoria no volátil puede comprender adicionalmente o como alternativa una EEPROM, memoria flash o similares. Las memorias pueden almacenar cualquiera de un número de aplicaciones de software, instrucciones, piezas de información, y datos, usados por el terminal para implementar las funciones del terminal.

Como se ha explicado anteriormente, el terminal **10** puede operar o ejecutar de otra manera un número de diferentes aplicaciones de software incluyendo, por ejemplo, un explorador de WAP (Protocolo de Aplicación Inalámbrica), cliente de MMS, cliente de SMS, cliente de correo electrónico, motor de conectividad de OBEX (Intercambio de Objetos) de corto alcance, conectividad de PC, pilas de Bluetooth e IR, PIM (gestión de información personal) y aplicaciones de telefonía. En el terminal, tales aplicaciones se ejecutan normalmente en la parte superior de una plataforma de SO, tal como la proporcionada por la Tecnología Genérica (GT) de Symbian™ de arquitectura abierta. Como se explica a continuación, el sistema operativo comprende Symbian OS™. Debería entenderse, sin embargo,

que el sistema operativo puede comprender cualquiera de un número de otros sistemas operativos que pueden operar similares al SO de Symbian™ de acuerdo con realizaciones de la presente invención. Por ejemplo, en lugar del SO de Symbian™, las aplicaciones pueden operar en la parte superior de sistemas operativos tales como Linux, Windows® CE o Palm OS®.

Como se apreciará, la GT de Symbian™ es un núcleo común de las interfaces de programación de aplicaciones (API) y de la tecnología de sistema operativo de Symbian™. Contiene todo de las interfaces a las aplicaciones de interfaz de usuario (UI), bibliotecas de enlace dinámico (DLL), ejecutables (EXE) y controladores de dispositivo para controlar diversos dispositivos de hardware del terminal **10** incluyendo, por ejemplo, el timbre **34,** pantalla **38,** teclado numérico **40** y, según sea aplicable, el transceptor de RF **42,** transceptor de IR **44** y/o transceptor de Bluetooth **46.** Además, la GT de Symbian™ comunica con el software celular del núcleo del terminal a través de una arquitectura de mensajería bien definida y documentada.

Se hace ahora referencia a la Figura 2, que ilustra un diagrama de arquitectura de la GT del SO de Symbian™ que opera en la parte superior de un número de aplicaciones **54** (por ejemplo, aplicación 1, aplicación 2, aplicación 3, etc.) a bordo de un terminal basado en Symbian™ **10.** La GT de SO de Symbian™ se muestra en la parte superior del hardware de terminal (véase la Figura 1) y en el software de nivel inferior **56.** Como es bien conocido, la GT de SO de Symbian™ comprende un número de subsistemas que dividen conceptualmente las API y la tecnología de sistema operativo proporcionada por la GT de Symbian™ en aquellas de la funcionalidad relacionada. En este sentido, como se muestra, la GT de Symbian™ incluye un subsistema de base **58,** un subsistema de estructura de aplicación **60,** subsistema de infraestructura de comunicación **62,** subsistema de exploración **64,** subsistema de estructura de mensajería **66,** subsistema de Java™ **68,** subsistema de conectividad **70,** subsistema de estructura multimedia **72** y motores de aplicación **74.**

En resumen, el subsistema de base **58** de la arquitectura de GT de Symbian™ proporciona la estructura de programación para los otros componentes de la arquitectura de GT de Symbian™. En este sentido, el subsistema de base incluye el núcleo de sistema del SO, que emplea una arquitectura de micro-núcleo en la que un núcleo funcional mínimo está separado de una o más funcionalidades extendidas y/o componentes específicos de cliente. Además del micro-núcleo, el subsistema de base incluye una biblioteca de usuario, así como el servidor de ficheros que proporciona acceso compartido a sistemas de relleno.

El subsistema de estructura de aplicación **60** incluye las API de soporte intermedio para servicios y componentes tales como gestión de datos, gestión de texto, portapapeles e internacionalización de soporte de fichero de recursos y componentes de UI gráficos (GUI) de núcleo. Para implementar tales servicios, a continuación, el subsistema de estructura de aplicación incluye sub-sistemas tales como una arquitectura de aplicación, servidor de ventana, servidor de vista y estructura de componente de GUI.

El subsistema de infraestructura de comunicación **62** proporciona estructuras y servicios para comunicaciones e interconexión en red dentro de la arquitectura de GT de Symbian™. El subsistema de infraestructura de comunicación puede incluir o estar asociado de otra manera con servidores tales como un servidor de comunicaciones, servidor de conectores y servidor de telefonía. Los servidores de comunicaciones, conectores y telefonía, a su vez, proporcionan soporte para un número de tecnologías y protocolos de comunicación incluyendo, por ejemplo, el Protocolo de Control de Transmisión/Protocolo de Internet (TCP/IP), GSM, CDMA, el protocolo de aplicación inalámbrica (WAP), así como comunicaciones en serie, y comunicaciones de corto alcance tales como de infrarrojos y Bluetooth.

El subsistema de exploración **64** de la arquitectura de GT de Symbian™ incluye exploradores para comunicar de acuerdo con el Lenguaje de Marcas Inalámbrico (WML) para servicios WAP, y el Lenguaje de Marcas de Hipertexto (HTML) para servicios de la Red Informática Mundial (WWW). El subsistema de estructura de mensajería 66 está basado en un servidor de mensajería, y proporciona soporte para enviar y recibir un número de diferentes tipos de mensajes incluyendo, por ejemplo, mensajes de texto, mensajes de SMS, mensajes del Servicio de Mensajería Mejorado (EMS), mensajes de MMS y correo electrónico.

El subsistema de Java™ **68** de la arquitectura de GT de Symbian™ proporciona soporte para servicios Java™ en la estructura de Java™ 2 Edición Micro (J2ME). El subsistema de conectividad **70** proporciona conectividad de interconexión en red de área personal (PAN) mediante un número de diferentes medios de comunicación incluyendo, por ejemplo, medios Bluetooth, Bus Serie Universal (USB) en serie, infrarrojos y Ethernet. El subsistema de estructura multimedia **72** incluye un servidor de medios, y proporciona un número de capacidades multimedia incluyendo, por ejemplo, reproducción y grabación de audio y vídeo, funcionalidad de envío por flujo continuo de audio y funcionalidad de formación de imágenes. Y los motores de aplicación **74** proporcionan la funcionalidad principal para aplicaciones Symbian™ convencionales. En este sentido, los motores de aplicación pueden incluir agenda (planificación), tareas pendientes, contactos, alarma y servidores mundiales, así como un motor de ayuda.

Como se apreciará, entonces, los sistemas operativos tales como el SO de Symbian™ incluyen una colección extensiva de bibliotecas o DLL para implementar un número de diferentes normas que incluyen, normas de interconexión en red, comunicación, mensajería, exploración telefonía y multimedia. El acceso a los servicios y

recursos del sistema operativo, sin embargo, puede implementarse por proveedores de servicio a través de una estructura cliente-servidor. En una estructura de este tipo, un programa cliente (por ejemplo, la aplicación **54)** solicita y recibe servicios proporcionados por un servidor (por ejemplo, servidor de ficheros, servidor de ventanas, servidor de vista, servidor de comunicaciones, servidor de conector y servidor de telefonía, servidor de mensajería, servidor de medios, servidor de agenda (planificación), servidor de tareas pendientes, servidor de contactos, servidor de alarma, servidor mundial, etc.). Las solicitudes se reciben por medio de una API definida por el servidor para recibir tales solicitudes. El cliente y servidor comunican de acuerdo con un protocolo de paso de mensaje, donde la comunicación se media por el núcleo de sistema de SO del subsistema de base **58.**

Para solicitar/proporcionar servicios, un cliente y servidor comunican por medio de un canal de comunicación, a menudo denominado como una sesión. En este sentido, un cliente envía un mensaje a un servidor a través de una sesión para solicitar un servicio proporcionado por el servidor, incluyendo el mensaje un código de tipo de solicitud de 32 bits y hasta cuatro parámetros de 32 bits, por ejemplo. El servidor puede a continuación dar servicio a la solicitud y devolver un resultado, tal como un resultado de 32 bits, al cliente. En diversos casos, sin embargo, puede requerirse que el servidor envíe y/o reciba datos adicionales para dar servicio a la solicitud. En tales casos, el servidor puede realizar una o más funciones de lectura y/o escritura de acuerdo con servicios de transferencia de datos inter-hilo antes de devolver el resultado para completar el servicio de la solicitud.

Como se explica en la sección de antecedentes, diversas aplicaciones de software **54** pueden ser maliciosas en apariencia de virus, caballos de Troya o algún otro elemento introducido por un pirata para abusar del terminal **10** o para obtener de otra manera acceso a la funcionalidad del terminal. Para ilustrar de manera más evidente esto, considérense dos aplicaciones, en concreto la aplicación 1 (app1) **54a** y la aplicación 2 (app2) **54b,** como se ilustra en la Figura 3. Como se muestra, las dos aplicaciones operan en la parte superior de una plataforma de SO (por ejemplo, Symbian OS™), y reciben servicios proporcionados por uno o más de los servidores, en concreto el servidor A **76a,** servidor B **76b** y/o servidor C **76c.** Y como se ha explicado anteriormente con respecto al Symbian OS™, la comunicación entre las aplicaciones y los servidores pasa a través de las respectivas API **78** de la plataforma de SO, y se media por el núcleo **80** del SO.

En el ejemplo mostrado en la Figura 3, también considérese que la aplicación 1 **54a** está operando de una manera apropiada, intentando leer y/o escribir ficheros en memoria (por ejemplo, memoria no volátil **52)** del terminal **10,** tal como en una base de datos de contactos o libreta de direcciones, como se representa por la sesión **82.** La aplicación 2 **54b,** por otra parte, es una aplicación maliciosa, que el usuario de terminal ha descargado previamente en el terminal. Cuando se lanza para operación en el terminal, la aplicación maliciosa 2 puede realizar una o más funciones inapropiadas. Por ejemplo, la aplicación maliciosa 2 puede leer, escribir y/o borrar ficheros a y/o desde memoria del terminal, y/o transferir uno o más ficheros a un dispositivo remoto mediante una técnica de mensajería (por ejemplo, mensajería de texto, SMS, EMS, MMS, correo electrónico, etc.), como se representa por la sesión **84.** En este escenario, la aplicación 2 puede comprometer la confianza y privacidad del usuario realizando funciones indeseables, que pueden dar como resultado pérdida personal y/o financiera para el usuario de terminal.

Para superar las desventajas asociadas al escenario de la Figura 3 (es decir, programas maliciosos que obtienen servicios de la plataforma de SO del terminal **10),** las realizaciones de la presente invención proporcionan una estructura de seguridad mediante la cual cada aplicación **54** instalada o cargada de otra manera en el terminal está asociada a un registro de permiso que identifica uno o más servicios que la aplicación está permitida a recibir desde la plataforma de SO. A continuación, cuando una aplicación solicita un servicio proporcionado por la plataforma de SO, puede comprobarse el registro de permiso para determinar si la aplicación está autorizada a recibir el servicio solicitado. Si la aplicación está autorizada, la plataforma de SO puede continuar para proporcionar el servicio solicitado a la aplicación. De otra manera, puede evitarse que la aplicación reciba el servicio solicitado.

Se hace referencia ahora a las Figuras 4, 5, y 6a y 6b, que ilustran las realizaciones de la presente invención. Más particularmente, la Figura 4 es un diagrama de bloques funcional de las aplicaciones que operan en la parte superior de una plataforma de SO, de acuerdo con una realización de la presente invención. Las Figuras 5 y 6a y 6b son diagramas de flujo que incluyen diversas etapas al validar una aplicación de software **54,** también de acuerdo con una realización de la presente invención.

Como se muestra en la Figura 4 y se explica a continuación, en contraste al diagrama de bloques de un terminal convencional como se muestra en la Figura 3, el terminal **10** de las realizaciones de la presente invención incluye adicionalmente una aplicación de instalador **85** que, como parte de la plataforma de SO, está adaptada para validar aplicaciones recibidas por el terminal, y si se validan, instalar tales aplicaciones en el terminal, y crear registros de permiso para tales aplicaciones. En este sentido, el terminal de las realizaciones de la presente invención también incluye una base de datos de políticas (DB) **84** para almacenar los registros de permisos creados por el terminal. Además, el terminal incluye una aplicación de gestión **86** para confirmar, añadir, borrar, anular o modificar de otra manera uno o más permisos en los registros de permisos de una o más aplicaciones. Además, el terminal incluye un módulo de validación **88,** integrado o asociado de otra manera a la o las API **78,** para determinar si las aplicaciones que solicitan servicios proporcionados por la plataforma de SO están autorizadas a recibir estos servicios, determinando el módulo de validación la autoridad de las aplicaciones basándose en respectivos registros de permiso en la base de datos de políticas.

Como se muestra en el bloque **90** de la Figura 5, un método de validación de software de una aplicación de software **54** incluye el terminal **10** que recibe una aplicación de software, tal como la aplicación 3 **54c** como se muestra en la Figura 4. El terminal puede recibir la aplicación de software en cualquiera de un número de diferentes maneras. Por ejemplo, el terminal puede descargar la aplicación de software desde un servidor de origen **24** a través de la Internet **22** y una o más redes celulares o móviles (véase la Figura 1). Independientemente de cómo el terminal recibe la aplicación de software, la aplicación de software puede almacenarse en memoria del terminal (por ejemplo, la SIM **48,** memoria volátil **50,** memoria no volátil **52,** etc.). A continuación, después de recibir y almacenar la aplicación de software, se inicia la instalación de la aplicación de software para operación en el terminal, tal como dirigiendo un usuario del terminal la aplicación de instalador **85** para instalar la aplicación de software, como se muestra en el bloque **92.**

Después de que se inicia la instalación de la aplicación de software **54**, pero antes de que se instale la aplicación de software, la aplicación de instalador **85** intenta verificar la autenticidad de la aplicación de software, como se muestra en los bloques **94** y **96.** Por ejemplo, la aplicación de instalador puede verificar la autenticidad del origen (por ejemplo, un servidor de origen **24)** de la aplicación de software para verificar de esta manera la autenticidad de la aplicación de software. En este sentido, la autenticidad de la aplicación de software puede verificarse en cualquiera de un número de diferentes maneras, tal como comprobando una suma de comprobación criptográfica (por ejemplo, firma de Guarda de Privacidad de GNU (GPG)). Si la aplicación de instalador falla al verificar la autenticidad de la aplicación de software, la aplicación de instalador puede evitar la instalación de la aplicación de software.

Si la aplicación de instalador **85** verifica la autenticidad de la aplicación de software **54,** la aplicación de instalador instala la aplicación de software para operación en el terminal, como se muestra en el bloque **98.** También después de verificar la autenticidad de la aplicación de software, la aplicación de instalador crea un registro de permiso para la aplicación de software, como se muestra en el bloque **100.** A continuación, como se ilustra en el bloque **102,** el registro de permiso puede a continuación almacenarse como una entrada en una base de datos de políticas **84** que reside dentro del espacio de núcleo de la plataforma de SO, como se representa por la sesión mostrada como **104** y **108.** En este sentido, la base de datos de políticas puede tener el acceso restringido únicamente a aplicaciones especificadas, que pueden estar autorizadas a leer y/o escribir en la base de datos de políticas.

La aplicación de instalador **85** puede crear el registro de permiso, o más particularmente determinar los permisos incluidos en el registro de permiso, en cualquiera de un número de diferentes maneras. Por ejemplo, la aplicación de instalador puede configurarse para incluir uno o más permisos por defecto en cada registro de permiso, o en cada registro de permiso de uno o más tipos particulares de aplicaciones de software **54.** Adicionalmente o como alternativa, por ejemplo, la aplicación de instalador puede recibir uno o más permisos desde la misma aplicación de software, tal como en un fichero "permissions.ini". En tales casos, la aplicación de instalador puede solicitar, tal como desde un usuario de terminal, confirmación de uno o más de los permisos recibidos antes de insertar estos permisos en el registro de permiso. Por ejemplo, suponiendo que la aplicación de software incluye uno o más permisos que autorizan a la aplicación de software a realizar una o más operaciones usando interfaces de E/S (por ejemplo, transmisor **26,** receptor **28,** transceptor de RF **42,** transceptor de IR **44,** transceptor de Bluetooth **46,** etc.) del terminal **10.** En un ejemplo de este tipo, la aplicación de instalador puede notificar al usuario de terminal de uno o más de los permisos que autorizan el uso de las interfaces de E/S del terminal y solicitar que el usuario de terminal confirme o deniegue aquellos permisos, tal como mediante una aplicación de gestión **86** como se explica a continuación.

Además, por ejemplo, el usuario de terminal puede operar una aplicación de gestión **86** antes, durante o después de que la aplicación de instalador **85** instale la aplicación de software **54** para confirmar, añadir, borrar, anular o modificar de otra manera uno o más permisos en los registros de permisos de una o más aplicaciones de software, según se representa por la sesión mostrada como **106** y **108.** En este sentido, además de confirmar permisos proporcionados por una aplicación de software que se está instalando, el usuario de terminal puede operar la aplicación de gestión para establecer, resetear o modificar de otra manera uno o más permisos por defecto incluidos en uno o más registros de permiso. También, por ejemplo, el usuario de terminal puede operar la aplicación de gestión para anular o modificar de otra manera permisos proporcionados por la aplicación de software, o establecerse como permisos por defecto. Además, por ejemplo, el usuario de terminal puede operar la aplicación de gestión para añadir y/o borrar permisos desde los permisos por defecto, o desde uno o más registros de permiso, tal como permisos relacionados con conectores y/o control de acceso de E/S.

Como se ha indicado anteriormente, el registro de permiso identifica directa o indirectamente uno o más servicios a los que la aplicación está permitida a recibir desde la plataforma de SO, incluyendo operaciones que la aplicación de software **54** está autorizada a realizar con respecto a la plataforma de SO. En este sentido, el registro de permiso puede identificar uno o más permisos asociados a la respectiva aplicación de software con respecto a la plataforma de SO del terminal **10.** Por ejemplo, un registro de permiso para una aplicación de software (por ejemplo, la aplicación 3 54c) puede identificar uno o más ficheros que la respectiva aplicación de software está autorizada a leer, operaciones de entrada/salida (E/S) que la aplicación de software puede realizar y/o qué comunicación inter-proceso (IPC) puede utilizar la aplicación de software para comunicar con uno o más procesos, y uno o más tipos de ficheros temporales que la aplicación de software está autorizada a crear, así como cuándo y dónde la aplicación de software está autorizada a crear los ficheros temporales. Más particularmente, por ejemplo, un registro de permiso

para la aplicación 3 puede leerse como sigue:

```
Nombre de programa - Aplicación 3
Acceso de fichero:
   Contact.db (R+) - No crear, borrar;
   Temp.txt (R, W) - Crear, borrar
Acceso de consola: Sí
```

En el fichero de permiso anterior, la aplicación 3 tiene acceso de fichero a la base de datos de contactos (contact.db) en memoria (por ejemplo, memoria no volátil **52**) del terminal **10,** así como acceso para crear ficheros de texto temporales (temp.txt). Más particularmente, la aplicación 3 tiene acceso de sólo lectura (R+) a la base de datos de contactos sin autorización a crear o borrar entradas en la base de datos de contactos, pero tiene el acceso de lectura/escritura (R, W) para crear ficheros de texto temporales con autorización a crear y borrar ficheros de texto temporales. Adicionalmente, además del acceso de fichero, la aplicación 3 tiene autorización a acceder a la consola del terminal (por ejemplo, pantalla **38,** teclado numérico **40,** etc.).

Además de uno o más permisos, el registro de permiso puede incluir adicionalmente una o más piezas de información adicionales relacionadas con la respectiva aplicación de software **54.** Por ejemplo, el registro de permiso puede identificar el origen (por ejemplo, un servidor de origen **24**) de la aplicación de software, así como cuándo recibió el terminal **10** la aplicación de software desde el origen, y/o dónde se almacena en memoria (por ejemplo, memoria no volátil **52)** la aplicación de software. También, por ejemplo, el registro de permiso puede almacenar, para la aplicación de software, una firma que puede generarse o recibirse de otra manera por la aplicación de instalador **85.** En este sentido, la firma puede comprender cualquiera de un número de diferentes valores, cadenas o similares, tal como una función de troceo del nombre de fichero de la aplicación de software, mediante la cual puede verificarse la integridad de la aplicación de software, como se explica a continuación.

En uno o más casos después de instalar la aplicación de software **54,** y crear/almacenar un registro de permiso para la aplicación de software, la aplicación de software puede cargarse o ejecutarse de otra manera para operación en el terminal **10.** Como alternativa, en diversos casos una aplicación puede configurarse para cargarse o ejecutarse de otra manera para operación en el terminal sin instalarse en primer lugar de manera separada. Incluso en tales casos, sin embargo, la aplicación puede procesarse por la aplicación de instalador **85,** tal como para crear y almacenar un registro de permiso para la aplicación, antes de cargar o ejecutar de otra manera en primer lugar la aplicación para operación en el terminal. En general, a continuación, después de que la aplicación está configurada para cargarse o ejecutarse de otra manera para operación en el terminal, en uno o más casos, un cargador de programa incluido en o asociado de otra manera a la plataforma de SO inicia o está dirigido para cargar la aplicación de software para operación, como se muestra en el bloque **110** de la Figura 6a. A continuación, el cargador de programa intenta verificar la integridad de la aplicación de software, como se muestra en los bloques **112** y **114.**

El cargador de programa puede verificar la integridad de la aplicación de software **66** en cualquiera de un número de diferentes maneras para comprobar que la aplicación de software está intacta y no se ha modificado de otra manera en la memoria (por ejemplo, memoria no volátil **52)** del terminal. Por ejemplo, el cargador de programa puede verificar la integridad de la aplicación de software generando una firma de verificación (por ejemplo, función de troceo del nombre de fichero) para la aplicación de software. La firma de verificación puede a continuación compararse con la firma en el registro de permiso de la aplicación de software almacenada en la base de datos de políticas **84.** Si la firma de verificación coincide sustancialmente, sino completamente, con la firma en el registro de permiso, el cargador de programa verifica la integridad de la aplicación de software. De otra manera, el cargador de programa ha fallado al verificar la integridad de la aplicación de software y puede evitar que la aplicación de software se cargue para operación.

Si el cargador de programa verifica la integridad de la aplicación de software **54,** el cargador de programa carga la aplicación de software para operación en el terminal **10,** como se muestra en el bloque **116.** Posteriormente, en uno o más casos durante la operación de la aplicación de software, la aplicación de software puede solicitar uno o más servicios proporcionados por la plataforma de SO, los servicios en ocasiones se proporcionan por un servidor **76** y se median por el núcleo **80** del SO. En tales casos, como se ilustra en el bloque **118** de la Figura 6b, la aplicación envía una solicitud de servicio a un respectivo servidor por medio de una API **78** definida por el servidor para recibir tales solicitudes, como se representa por las sesiones **128** y **130** con respecto a la aplicación 1 **54a** y la aplicación 2 **54b,** respectivamente.

Tras la recepción de la solicitud por la API **78,** pero antes de pasar la solicitud al respectivo servidor, la API dirige el módulo de validación **88** para intentar verificar la autoridad de la aplicación solicitante **54** para recibir el servicio solicitado, como se ilustra en los bloques **120** y **122.** En este sentido, el módulo de validación carga el registro de permiso para la aplicación solicitante desde la base de datos de políticas **84** en la memoria de sistema (por ejemplo, la memoria volátil **50)** del terminal **10.** A partir del registro de permiso, a continuación, el módulo de validación determina si la aplicación solicitante está autorizada que va a recibir el servicio solicitado basándose en los permisos incluidos en el registro de permiso. Por ejemplo, suponiendo que una aplicación solicita leer desde la base de datos de contactos. En un ejemplo de este tipo, el módulo de validación puede determinar si la aplicación solicitante está

autorizada a leer desde la base de datos de contactos buscando el respectivo registro de permiso para un permiso que autoriza a que la aplicación lea desde la base de datos de contactos (por ejemplo, acceso de fichero: Contact.db (R+)).

Si el módulo de validación falla al verificar la autoridad de la aplicación al recibir el servicio solicitado, el módulo de validación evita que la aplicación reciba el servicio solicitado, tal como evitando que la solicitud de servicio alcance el respectivo servidor. Adicionalmente, si se desea así, puede generarse un registro cronológico y almacenarse en memoria (por ejemplo, memoria no volátil 52) del terminal, tal como el núcleo 80 del SO. En este sentido, el registro cronológico puede identificar un número de diferentes piezas de información tal como, por ejemplo, la aplicación solicitante, el servicio solicitado y/o la hora/fecha de la aplicación solicitante que solicitó el servicio.

Si el módulo de validación **88** verifica la autoridad de la aplicación **54** que va a recibir el servicio solicitado, por otra parte, el módulo de validación pasa, o dirige la respectiva API **78** para que pase, la solicitud de servicio al respectivo servidor **76.** Posteriormente, como se ilustra en el bloque **124,** si el servidor está permitido a recibir, y recibe, la solicitud de servicio, el servidor proporciona el servicio solicitado a la aplicación, también representado por las sesiones **128** y **130** con respecto a la aplicación 1 **54a** y la aplicación 2 **54b,** respectivamente.

En uno o más casos durante la operación de la aplicación, la misma aplicación **54** puede solicitar el mismo servicio o diferente servicio desde el mismo servidor o diferentes servidores, como se muestra en el bloque **126.** Para cada servicio, a continuación, la API **78** recibe la solicitud, dirige el módulo de validación **88** para intentar verificar la aplicación para que reciba el servicio solicitado, y si se verifica, pasa la solicitud al respectivo servidor de manera que el respectivo servidor proporciona el servicio solicitado. Para mejorar la eficacia al verificar la autoridad de, y proporcionar, servicios posteriores a una aplicación, el registro de permiso de la aplicación solicitante puede mantenerse en memoria de sistema (por ejemplo, memoria volátil **50)** del terminal **10,** tal como hasta que se cierre la respectiva aplicación, como se representa por la sesión **132.** Por lo tanto, el módulo de validación puede verificar la autoridad de la aplicación basándose en el registro de permiso mantenido en memoria de sistema para solicitudes posteriores, a diferencia de cargar el registro de permiso desde la base de datos de políticas **84.**

Como se ha explicado anteriormente, una aplicación **54** solicita un servicio desde un servidor **76** adaptado para proporcionar el servicio solicitado. En uno o más casos, sin embargo, un servidor de base (por ejemplo, el servidor A **76a)** que recibe la solicitud de servicio desde una aplicación puede solicitar, a su vez, un servicio desde uno o más servidores posteriores (por ejemplo, el servidor B **76b)** en nombre de la aplicación solicitante. El servicio solicitado por el servidor de base en nombre de la aplicación puede proporcionarse a continuación a la aplicación desde servidores posteriores. En tales casos, la autoridad de la aplicación para recibir el servicio solicitado puede verificarse con respecto al servicio solicitado desde el servidor de base (véanse los bloques **120** y **122).** En respuesta a la recepción de la solicitud de servicio, el servidor de base puede solicitar un servicio desde un servidor posterior en nombre de la aplicación solicitante, enviándose la solicitud desde el servidor de base a través de una sesión mediante una respectiva API **78** definida por el servidor posterior, como se representa por la sesión **134** entre el servidor A (es decir, servidor de base) y el servidor B (es decir, servidor posterior) en la Figura 4.

Cuando se envía una solicitud de servicio desde un servidor **76** u otro componente confiable, tal como otro componente de la plataforma de SO, mediante una API **78,** la API puede configurarse para pasar la solicitud a un respectivo servidor sin dirigir el módulo de validación **88** para intentar verificar el servidor para recibir el servicio solicitado, incluso considerando que el servicio se proporcionará a la aplicación solicitante **54.** Como alternativa, cada componente de la plataforma de SO puede asociarse también a un registro de permiso. En tales casos, la autoridad del servidor puede verificarse, tal como de la misma manera que las aplicaciones, aunque los componentes de la plataforma de SO pueden considerarse más normalmente componentes confiables con acceso no restringido a servicios y recursos de la plataforma de SO. Independientemente de cómo se pase exactamente la solicitud desde el servidor de base al servidor posterior, el servidor posterior puede proporcionar posteriormente el servicio solicitado a la aplicación solicitante, a diferencia del servidor de base solicitante (aunque el servicio puede proporcionarse a la aplicación solicitante de vuelta a través del servidor de base).

Para ilustrar adicionalmente beneficios de las realizaciones de la presente invención, de nuevo considérense dos aplicaciones, en concreto la aplicación 1 (app1) **54a** y la aplicación 2 (app2) **54b**, como se ilustra en Figura 4. Como se muestra, como en la Figura 3, las dos aplicaciones de la Figura 4 operan en la parte superior de una plataforma de SO (por ejemplo, Symbian OS™), y reciben servicios proporcionados por uno o más de los servidores, en concreto el servidor A 76a, el servidor B **76b** y/o el servidor C **76c**. Y como se ha explicado anteriormente con respecto a Symbian OS™, la comunicación entre las aplicaciones y los servidores pasa a través de las respectivas API **78** de la plataforma de SO, y se media por el núcleo **80** del SO.

En el ejemplo mostrado en la Figura 4, también considérese que la aplicación 1 **54a** está operando de una manera apropiada, intentando leer y/o escribir ficheros en memoria (por ejemplo, memoria no volátil **52**) del terminal **10**, tal como en una base de datos de contactos o libreta de direcciones, como se representa por la sesión **128**. La aplicación 2 **54b**, por otra parte, es una aplicación maliciosa que también intenta leer y/o escribir ficheros en memoria, como se representa por la sesión **130**, que el usuario de terminal descargó previamente en el terminal. Durante la instalación, la aplicación de instalador **85** es probable que verifique la autenticidad de la aplicación 1. Pero

siendo una aplicación maliciosa, la aplicación de instalador puede fallar al verificar la autenticidad de la aplicación 2, y por lo tanto evitar la instalación de la aplicación 2. Considérese, sin embargo, que la aplicación de instalador también verifica satisfactoriamente la autenticidad de la aplicación 2. En este caso, la aplicación de instalador instala ambas aplicaciones, y crea y almacena un registro de permiso para ambas aplicaciones.

Ya que la aplicación de instalador **85** crea el registro de permiso para la aplicación maliciosa 2 **54b**, el registro de permiso de la aplicación 2 puede incluir únicamente los permisos por defecto, que pueden seleccionarse para incluir únicamente servicios que no pueden abusar del terminal de cualquier manera significativa para el usuario de terminal. También, durante la creación del registro de permiso, la aplicación 2 puede solicitar la autorización para recibir uno o más servicios que facilitan operación maliciosa de la aplicación 2. Como con otros permisos recibidos o proporcionados de otra manera por las aplicaciones, la aplicación de instalador puede solicitar que el usuario de terminal confirme los permisos recibidos antes de insertar estos permisos en el registro de permiso. El usuario de terminal puede reconocer a continuación la aplicación 2 como una aplicación maliciosa, o reconocer que la aplicación 2 no necesita los servicios para los que se solicita autorización, y denegar autorización para estos servicios.

Después de que la aplicación de instalador **85** instala la aplicación 1 **54a** y la aplicación 2 **54b**, cualquier aplicación **54** puede operarse en el terminal **10**. Durante la operación, la aplicación 1 es probable que solicite servicios que la aplicación 1 está autorizada a recibir, como se refleja en el registro de permiso para la aplicación 1, tal como solicitando/recibiendo un servicio a través de la sesión **128**. La aplicación maliciosa 2, por otra parte, puede considerarse benigna por la aplicación del registro de permiso de la aplicación 2 antes de que la plataforma de SO proporcione algún servicio a la aplicación 2. En este sentido, si la aplicación 2 solicita un servicio no incluido en los permisos del respectivo registro de permiso, tal como solicitar acceso de escritura a ficheros de sistema a través de la sesión **130**, el módulo de validación **88** evita que la aplicación 2 reciba el servicio solicitado. La aplicación 2 puede estar autorizada a recibir otros servicios por el registro de permiso, pero normalmente únicamente aquellos servicios que no puedan abusar del terminal de cualquier manera significativa para el usuario de terminal, como se ha indicado anteriormente.

De acuerdo con un aspecto de la presente invención, todo o una porción del sistema de la presente invención, tal como todo o porciones del terminal **10**, operan en general bajo el control de uno o más productos de programa informático (por ejemplo, aplicaciones de software **54**, plataforma de SO que incluye el servidor o servidores **76**, la o las API **78**, núcleo **80**, aplicación de instalador **85**, base de datos de políticas **84**, aplicación de gestión **86** y módulo de validación **88**, etc.). El producto de programa informático para realizar los métodos de las realizaciones de la presente invención incluye un medio de almacenamiento legible por ordenador, tal como el medio de almacenamiento no volátil, y porciones de código de programa legible por ordenador, tal como una serie de instrucciones informáticas, incorporadas en el medio de almacenamiento legible por ordenador.

En este sentido, las Figuras 5, 6a y 6b son diagramas de flujo de los métodos, sistemas y productos de programa de acuerdo con la invención. Se entenderá que cada bloque o etapa de los diagramas de flujo, y combinaciones de bloques en los diagramas de flujo, puede implementarse por instrucciones de programa informático. Estas instrucciones de programa informático pueden cargarse en un ordenador u otro aparato programable para producir una máquina, de manera que las instrucciones que se ejecutan en el ordenador u otro aparato programable crean medios para implementar las funciones especificadas en el bloque o bloques o etapa o etapas de los diagramas de flujo. Estas instrucciones de programa informático pueden almacenarse también en una memoria legible por ordenador que puede dirigir un ordenador u otro aparato programable para funcionar de una manera particular, de manera que las instrucciones almacenadas en la memoria legible por ordenador producen un artículo de fabricación que incluye medios de instrucción que implementan la función especificada en el bloque o bloques o etapa o etapas de los diagramas de flujo. Las instrucciones de programa informático pueden cargarse también en un ordenador u otro aparato programable para provocar que se realice una serie de etapas operacionales en el ordenador u otro aparato programable para producir un proceso implementado por ordenador de manera que las instrucciones que se ejecutan en el ordenador u otro aparato programable proporcionan etapas para implementar las funciones especificadas en el bloque o bloques o etapa o etapas de los diagramas de flujo.

Por consiguiente, los bloques o etapas de los diagramas de flujo soportan combinaciones de medios para realizar las funciones especificadas, combinaciones de etapas para realizar las funciones especificadas y medios de instrucción de programa para realizar las funciones especificadas. Se entenderá también que cada bloque o etapa de los diagramas de flujo, y combinaciones de bloques o etapas en los diagramas de flujo, puede implementarse por sistemas informáticos basados en hardware de fin especial que realizan las funciones o etapas especificadas, o combinaciones de hardware de fin especial e instrucciones informáticas.

Muchas modificaciones y otras realizaciones de la invención se le ocurrirán a un experto en la materia a la que pertenece esta invención que tiene el beneficio de las enseñanzas presentadas en las descripciones anteriores y los dibujos asociados. Por lo tanto, se ha de entender que la invención no ha de estar limitada a las realizaciones específicas desveladas y que se pretende que estén incluidas las modificaciones y otras realizaciones en el alcance de las reivindicaciones adjuntas. A pesar de que en el presente documento se emplean expresiones específicas, las mismas se usan solo en un sentido genérico y descriptivo y no para fines de limitación.

REIVINDICACIONES

1. Un método que comprende un procesador:

recibir una aplicación de software por un aparato (90);
operar una plataforma de sistema operativo del aparato;
verificar una autenticidad de la aplicación de software (94, 96) y, únicamente si se verifica la autenticidad de la aplicación de software:

instalar la aplicación de software en el aparato (98) para operación por encima de la plataforma de sistema operativo;
crear un registro de permiso para la aplicación de software (100); y
almacenar el registro de permiso en una base de datos de políticas de la plataforma de sistema operativo (102), incluyendo el registro de permiso al menos un permiso que identifica al menos un servicio al que la aplicación de software está autorizada a recibir desde la plataforma de sistema operativo, siendo editable el registro de permiso por un usuario del aparato para indicar al menos un servicio que el usuario ha autorizado a la aplicación de software a recibir desde la plataforma de sistema operativo;

recibir una solicitud desde la aplicación de software para un servicio de la plataforma de sistema operativo (118), recibiéndose la solicitud después de que la aplicación de software se ejecuta para operación y mientras se opera;
determinar si la aplicación de software está autorizada a recibir el servicio solicitado basándose en el registro de permiso (120, 122);
proporcionar el servicio solicitado a la aplicación de software si la aplicación de software está autorizada a recibir el servicio solicitado (124); y
mantener el registro de permiso en una memoria volátil (50) hasta que se cierra la aplicación de software.

2. Un método de acuerdo con la reivindicación 1, en el que recibir una solicitud desde la aplicación de software comprende recibir una pluralidad de solicitudes desde la aplicación de software para al menos un servicio de la plataforma de SO,

en donde determinar si la aplicación de software está autorizada comprende determinar, para cada servicio solicitado, si la aplicación de software está autorizada a recibir el servicio solicitado basándose en el registro de permiso asociado.

3. Un método de acuerdo con la reivindicación 1, en el que verificar una autenticidad comprende verificar una autenticidad de la aplicación de software basándose en un origen de la aplicación de software, habiéndose recibido previamente la aplicación de software desde el origen.

4. Un método de acuerdo con la reivindicación 1, que comprende adicionalmente:

iniciar la carga de la aplicación de software;
verificar una integridad de la aplicación de software; y
cargar la aplicación de software para operación en el aparato si se verifica la integridad de la aplicación de software,
en donde iniciar la carga de la aplicación de software, verificar la integridad y cargar la aplicación de software tiene lugar antes de recibir una solicitud desde la aplicación de software para un servicio de la plataforma de sistema operativo.

5. Un método de acuerdo con la reivindicación 4, en el que el registro de permiso asociado a la aplicación de software incluye adicionalmente una firma asociada a la aplicación de software, y en el que verificar una integridad de la aplicación de software comprende:

generar una firma de verificación basándose en la aplicación de software;
comparar la firma de verificación con la firma en el registro de permiso asociado a la aplicación de software; y
verificar la integridad de la aplicación de software basándose en la comparación.

6. Un método de acuerdo con cualquier reivindicación anterior, en el que el aparato es un terminal móvil.

7. Un método de acuerdo con cualquier reivindicación anterior, en el que la base de datos de políticas está incluida en un núcleo (80) de la plataforma de sistema operativo (102).

8. Un medio de almacenamiento legible por ordenador que comprende instrucciones legibles por máquina que cuando se ejecutan por el aparato informático lo controlan para realizar el método de cualquiera de las reivindicaciones 1 a 7.

9. Aparato (10) que comprende:

medios para recibir una aplicación de software (54a, 54b);
medios para operar una plataforma de sistema operativo del aparato;
medios para verificar una autenticidad de la aplicación de software (54a, 54b) y, únicamente si se verifica la autenticidad de la aplicación de software (54a, 54b):

instalar la aplicación de software (54a, 54b) en el aparato (10) para operación por encima de la plataforma de sistema operativo;
crear un registro de permiso para la aplicación de software (54a, 54b); y
almacenar el registro de permiso en una base de datos de políticas (108) de la plataforma de sistema operativo, incluyendo el registro de permiso al menos un permiso que identifica al menos un servicio que la aplicación de software (54a, 54b) está autorizada a recibir desde la plataforma de sistema operativo, siendo editable el registro de permiso por un usuario del aparato (10) para indicar al menos un servicio que el usuario ha autorizado a la aplicación de software (54a, 54b) a recibir desde la plataforma de sistema operativo;

medios (78) para recibir una solicitud desde la aplicación de software para un servicio de la plataforma de sistema operativo, recibiéndose la solicitud después de que la aplicación de software (54a, 54b) se ejecuta para operación y mientras se opera;
medios (88) para determinar si la aplicación de software (54a, 54b) está autorizada a recibir el servicio solicitado basándose en el registro de permiso;
medios (76a, 76b, 76c) para proporcionar el servicio solicitado a la aplicación de software (54a, 54b) si la aplicación de software está autorizada a recibir el servicio solicitado; y
medios para mantener el registro de permiso en una memoria volátil (50) hasta que se cierra la aplicación de software.

10. El aparato de la reivindicación 9, en el que recibir una solicitud desde la aplicación de software comprende recibir una pluralidad de solicitudes desde la aplicación de software para al menos un servicio de la plataforma de SO,
en donde determinar si la aplicación de software está autorizada comprende determinar, para cada servicio solicitado, si la aplicación de software está autorizada a recibir el servicio solicitado basándose en el registro de permiso asociado.

11. El aparato de la reivindicación 9, en el que verificar una autenticidad comprende verificar una autenticidad de la aplicación de software basándose en un origen de la aplicación de software, habiéndose recibido previamente la aplicación de software desde el origen.

12. El aparato de la reivindicación 9, que comprende adicionalmente:

medios para iniciar la carga de la aplicación de software;
medios para verificar una integridad de la aplicación de software; y
medios para cargar la aplicación de software para operación en el aparato si se verifica la integridad de la aplicación de software,

en donde el aparato está configurado de manera que iniciar la carga de la aplicación de software, verificar la integridad y cargar la aplicación de software tienen lugar antes de recibir una solicitud desde la aplicación de software para un servicio de la plataforma de sistema operativo.

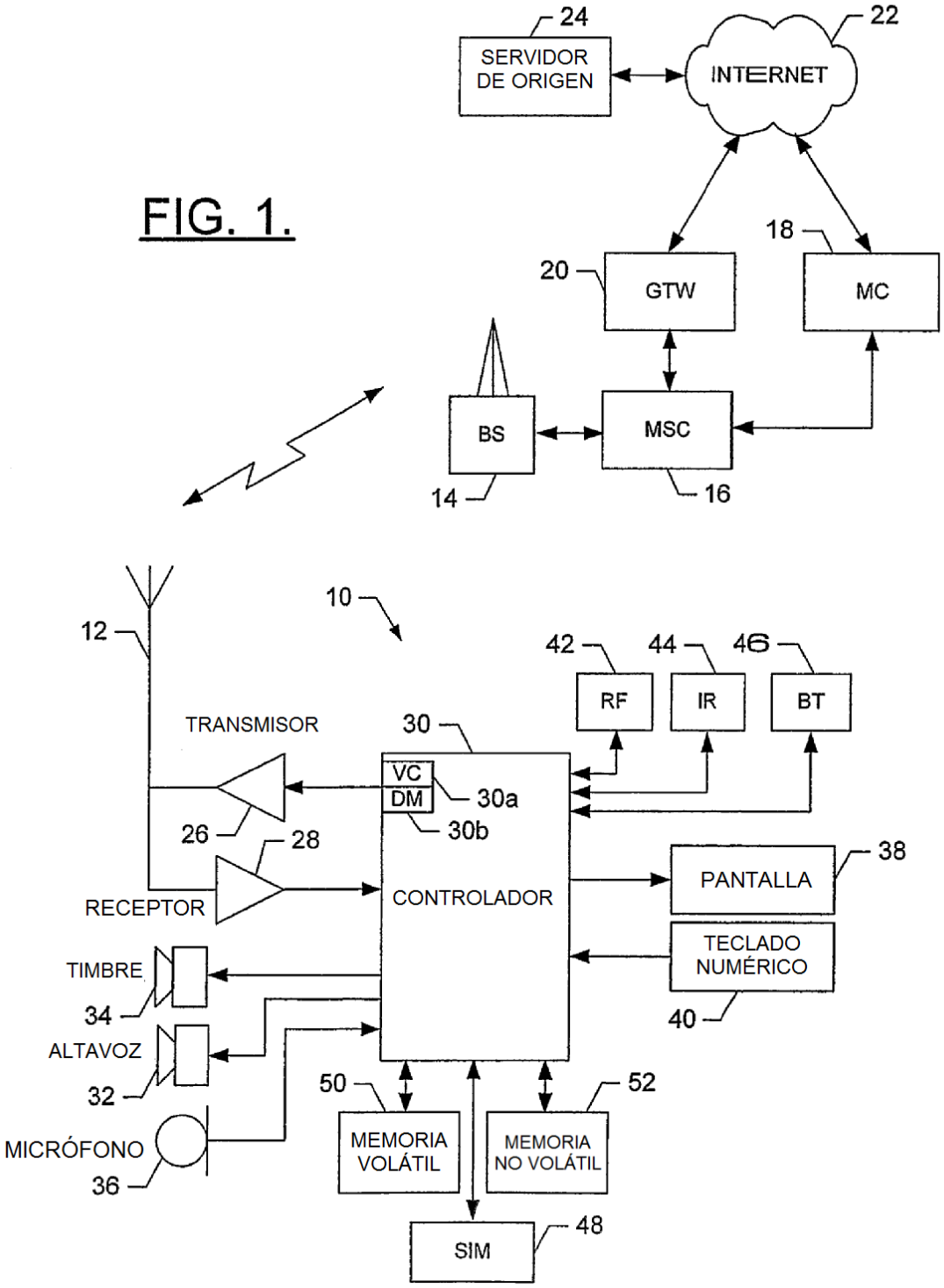
13. El aparato de la reivindicación 12, en el que el registro de permiso asociado a la aplicación de software incluye adicionalmente una firma asociada a la aplicación de software, y en el que verificar una integridad de la aplicación de software comprende:

generar una firma de verificación basándose en la aplicación de software;
comparar la firma de verificación con la firma en el registro de permiso asociado a la aplicación de software; y
verificar la integridad de la aplicación de software basándose en la comparación.

14. El aparato de cualquiera de las reivindicaciones 9 a 13, en donde el aparato es un terminal móvil.

15. El aparato de cualquiera de las reivindicaciones 9-14, en el que la base de datos de políticas está incluida en un núcleo (80) de la plataforma de sistema operativo (102).

FIG. 1.


24
SERVIDOR DE ORIGEN
22
INTERNET
18
20
GTW
MC
BS
MSC
14
16
10
12
42
44
46
RF
IR
BT
TRANSMISOR
30
VC
DM
30a
30b
26
28
RECEPTOR
CONTROLADOR
PANTALLA
38
TECLADO NUMÉRICO
40
TIMBRE
34
ALTAVOZ
32
MICRÓFONO
36
50
52
MEMORIA VOLÁTIL
MEMORIA NO VOLÁTIL
SIM
48

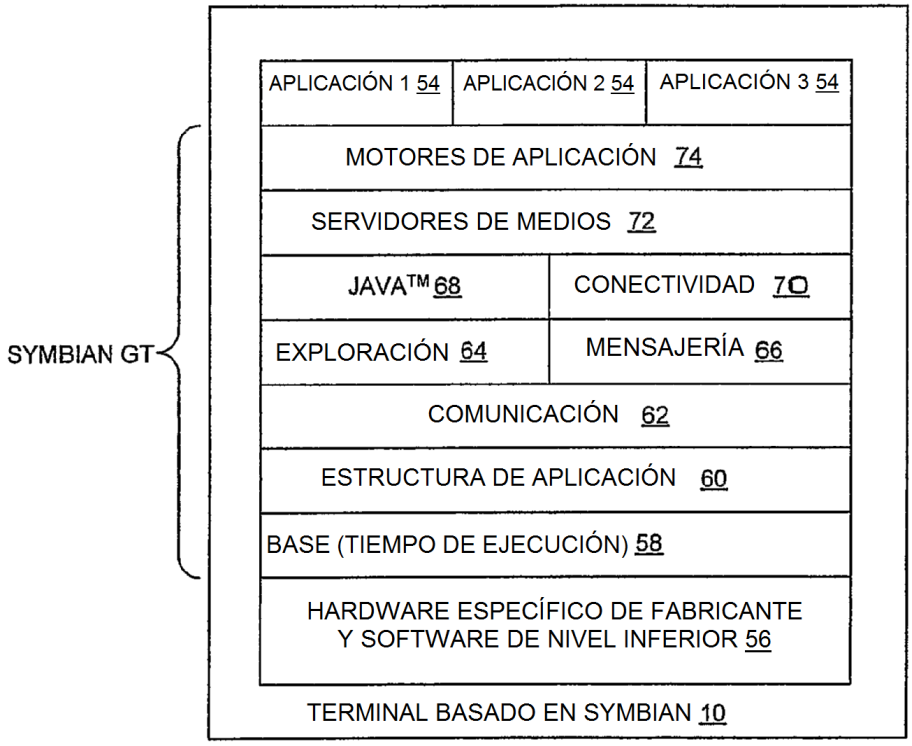

APLICACIÓN 1 54
APLICACIÓN 2 54
APLICACIÓN 3 54
MOTORES DE APLICACIÓN 74
SERVIDORES DE MEDIOS 72
JAVA™ 68
CONECTIVIDAD 70
EXPLORACIÓN 64
MENSAJERÍA 66
COMUNICACIÓN 62
ESTRUCTURA DE APLICACIÓN 60
BASE (TIEMPO DE EJECUCIÓN) 58
HARDWARE ESPECÍFICO DE FABRICANTE Y SOFTWARE DE NIVEL INFERIOR 56
SYMBIAN GT
TERMINAL BASADO EN SYMBIAN 10

FIG. 2.

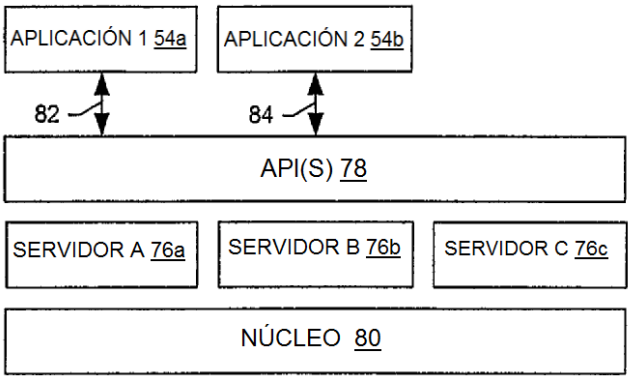


FIG. 3.
(TÉCNICA ANTERIOR)

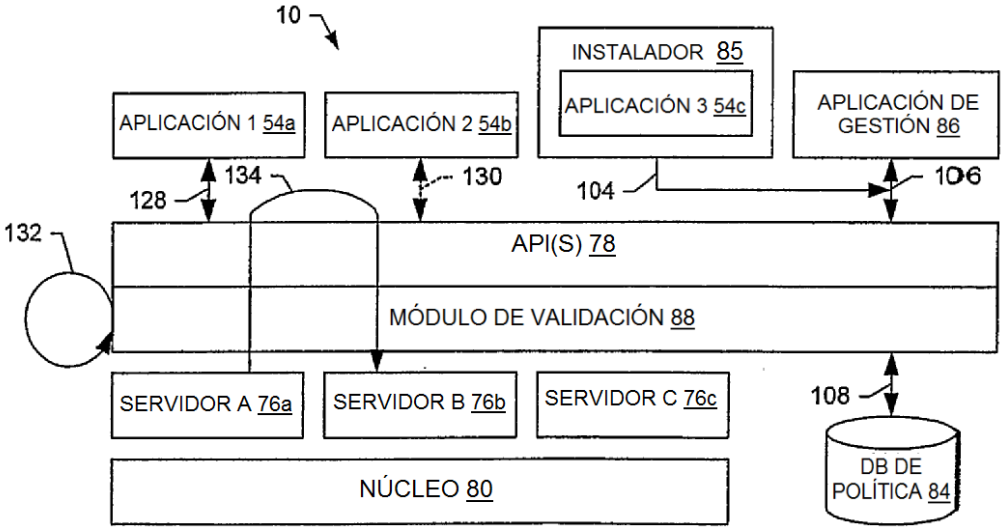


FIG. 4.

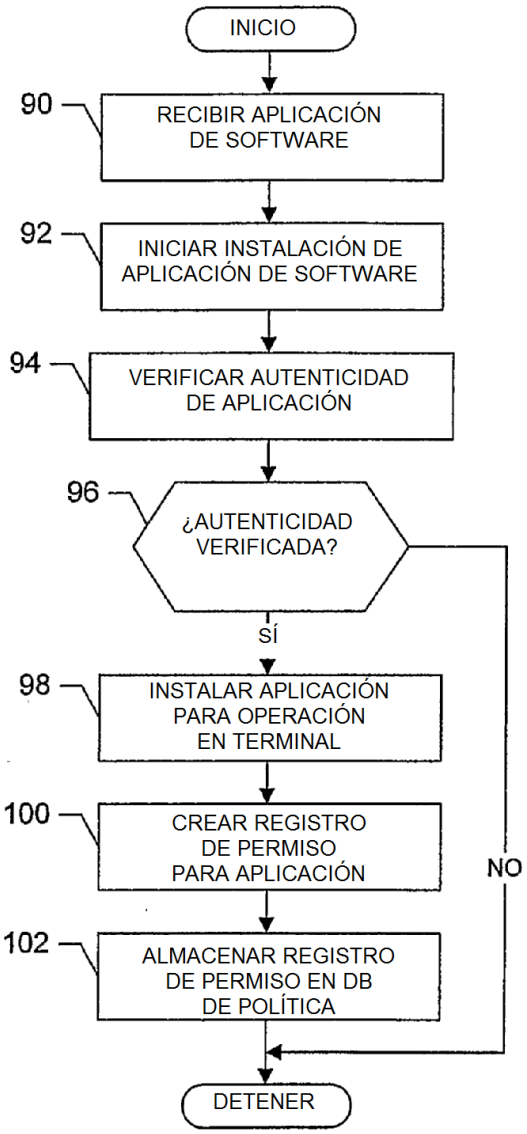

INICIO
90
RECIBIR APLICACIÓN DE SOFTWARE
92
INICIAR INSTALACIÓN DE APLICACIÓN DE SOFTWARE
94
VERIFICAR AUTENTICIDAD DE APLICACIÓN
96
¿AUTENTICIDAD VERIFICADA?
SÍ
NO
98
INSTALAR APLICACIÓN PARA OPERACIÓN EN TERMINAL
100
CREAR REGISTRO DE PERMISO PARA APLICACIÓN
102
ALMACENAR REGISTRO DE PERMISO EN DB DE POLÍTICA
DETENER

FIG. 5.

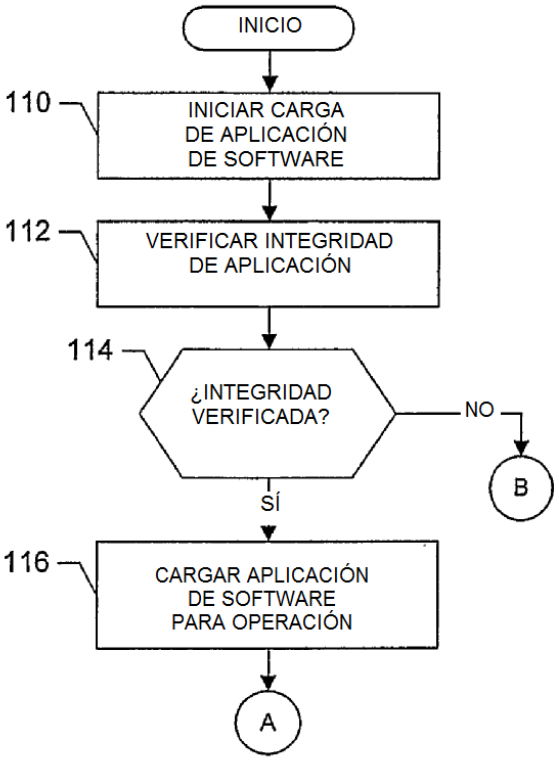

INICIO
110
INICIAR CARGA DE APLICACIÓN DE SOFTWARE
112
VERIFICAR INTEGRIDAD DE APLICACIÓN
114
¿INTEGRIDAD VERIFICADA?
NO
B
SÍ
116
CARGAR APLICACIÓN DE SOFTWARE PARA OPERACIÓN
A

FIG. 6a.

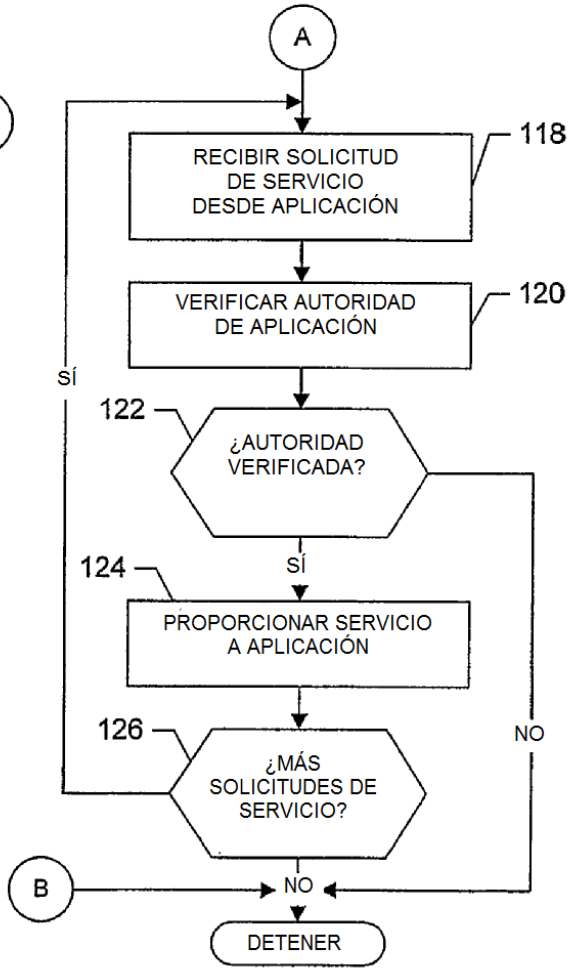

A
118
RECIBIR SOLICITUD DE SERVICIO DESDE APLICACIÓN
120
VERIFICAR AUTORIDAD DE APLICACIÓN
SÍ
122
¿AUTORIDAD VERIFICADA?
SÍ
124
PROPORCIONAR SERVICIO A APLICACIÓN
126
¿MÁS SOLICITUDES DE SERVICIO?
NO
B
NO
DETENER

FIG. 6b.