(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 362 358**

(51) Int. Cl.:
G06F 21/00 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Número de solicitud europea: **04007646 .5**
(96) Fecha de presentación : **30.03.2004**
(97) Número de publicación de la solicitud: **1465041**
(97) Fecha de publicación de la solicitud: **06.10.2004**

(54) Título: **Dispositivo de comunicación, procedimiento y programa para comprobar el permiso de ejecución de software.**

(30) Prioridad: **31.03.2003 JP 2003-96015**

(45) Fecha de publicación de la mención BOPI:
04.07.2011

(45) Fecha de la publicación del folleto de la patente:
04.07.2011

(73) Titular/es: **NTT DoCoMo, Inc.**
11-1, Nagatacho 2-Chome
Chiyoda-Ku, Tokyo 100-6150, JP

(72) Inventor/es: **Ichikawa, Yuichi;**
Naruse, Naoki;
Oi, Tatsuro;
Watanabe, Nobuyuki;
Hattori, Yasunori;
Takeshita, Masato;
Nishida, Masakazu;
Asai, Mao;
Tsuda, Masayuki;
Tomioka, Atsuki;
Yamada, Kazuhiro;
Kamiya, Dai;
Washio, Satoshi;
Yamane, Naoki y
Murakami, Keiichi

(74) Agente: **Carpintero López, Mario**

DESCRIPCIÓN

Dispositivo de comunicación, procedimiento y programa para comprobar el permiso de ejecución de software

Campo técnico

La presente invención se refiere a una técnica para garantizar la aplicación de los derechos de comportamiento.

Técnica antecedente

En el documento EP 1 132 796 A 1, se describe un código móvil y un procedimiento para la gestión de recursos con destino al código móvil. Un código móvil está vinculado a un certificado que incluye al menos una lista de requisitos de necesidades, RRL, que incluye aquellos recursos que el código móvil necesita para que puedan ser adecuadamente ejecutados más aquellos recursos que son conocidos a priori a los que puede accederse al ejecutar el código móvil. El certificado exclusivo contiene, así mismo, un emisor de la información del certificado que identifica la entidad del certificado, una información de la materia que identifica el código móvil al que el certificado se refiere, y una información del periodo de validez a partir del periodo de tiempo en el que el certificado es válido. Al descargar o cargar un código móvil, la RRL es transferida al usuario para informar al usuario de los requisitos de las necesidades del código móvil. Se proporciona un entorno de ejecución es proporcionado en un par de ejecución del usuario, definiendo el entorno de ejecución al menos la política de acceso a los recursos que será aplicada al código móvil en el momento de la ejecución.

Recientemente, han aparecido muchas funciones nuevas para su uso en estaciones móviles. Estas incluyen, por ejemplo, funciones para la descarga de software que contienen programas de aplicación escritos en lenguaje de programación Java (marca registrada) (el programa de aplicación será designado como “Java - APP”, y el software será designado como “Software Java - AP”, en lo sucesivo), a través de las redes de comunicación y, así mismo, funciona para ejecutar las Java - APPs.

El comportamiento de las aplicaciones, las cuales son grupos de funciones realizadas en estaciones móviles cuando son ejecutados los Java - APPs (la aplicación será en lo sucesivo designada como “Java - AP”) está generalmente sometida a una mayor restricción que la de los programas de aplicación nativos, programas de aplicación que son instalados en las estaciones móviles antes de su distribución a los usuarios; e incluyen aplicaciones de comunicaciones de correo electrónico, y similares (por ejemplo, remítase a la Patente japonesa abierta a inspección pública No. 10-254783).

De modo específico, en general, a las Java - APs no se les permite acceder a los datos confidenciales, como por ejemplo los datos de los directorios telefónicos almacenados en una estación móvil. La razón para restringir el comportamiento de las Java - APs de la forma indicada es para impedir las filtraciones o la falsificación de los datos confidenciales por las Java - APs las cuales se llevan a cabo por las Java - APPs que contienen un código doloso o errores de código.

Sin embargo, la imposición de una restricción del comportamiento escrito para todas las Java - APs hace imposible satisfacer las demandas de los usuarios que desean utilizar una diversidad de nuevas y útiles funciones de las Java - APs. De modo similar, dicha restricción generalizada hace, así mismo, imposible que los Proveedores de Contenidos (designados en lo sucesivo como “CPs”) proporcionen a los usuarios los Java - APPs que sean capaces de proporcionar una diversidad de dichas funciones.

En el caso de que pueda asegurarse la seguridad razonable (fiabilidad) de dichos Java- APPs es probable que los usuarios no pondrían obstáculos a que los Java - APs tuvieran la posibilidad de acceder a los datos personales almacenados en una estación móvil en la cual son ejecutadas. Así mismo, sería conveniente que los CPs fueran capaces de proporcionar a los usuarios unos Java - APPs más funcionales de los que actualmente proporcionan, y que pudieran proporcionar diversas funciones en la utilización de los datos personales almacenados en las estaciones móviles.

Para satisfacer las demandas expuestas con anterioridad, se han propuesto diferentes sistemas, como por ejemplo el esbozado a continuación. En este ejemplo, una portadora que proporciona servicios de comunicación a los usuarios de estaciones móviles garantiza los derechos de comportamiento con menores restricciones de las Java - APs, que las que normalmente se aplican; si la portadora informa a las estaciones móviles del tipo de derechos que han sido garantizadas a las Java - APs. La portadora sería una que fuera considerada normalmente como fiable en términos de seguridad por parte de los usuarios; de esta manera, en la presente exposición se utiliza la expresión “agencia fiable”. En el sistema en cuestión, la restricción del comportamiento de las Java - APs es controlada por una estación móvil sobre la base de los derechos informados a la estación móvil por parte de la agencia fiable.

Sin embargo este sistema presenta un problema. A saber, durante el uso, la agencia fiable informa a las estaciones móviles de aquellos derechos de comportamiento respecto de los cuales se ha producido una relajación de las restricciones para las Java - APs operables en las estaciones móviles. Por tanto, las estaciones móviles, de acuerdo con ello, establecen restricciones para el comportamiento de las Java - APs. Sin embargo, puede resultar necesario que la agencia fiable modifique la información acerca de dichos derechos de comportamiento. Esto sería el caso, por

ejemplo, en una situación en la que se ha comprendido de forma restrospectiva que una relajación de los derechos de comportamiento de una Java - AP está teniendo un efecto perjudicial en los usuarios. Dicho supuesto podría incluir aquél en el que una Java - AP leyera los datos personales almacenados en una memoria de una estación móvil y transmitiera los datos a un dispositivo externo sin el consentimiento del propietario. En este caso, sería necesario que la agencia fiable modificara de manera inmediata el comportamiento permisible de la Java - AP. Dicho claramente, habrá ocasiones en las que para garantizar la seguridad de los datos resulte necesario modificar (rápidamente) los derechos de comportamiento garantizados por una agencia fiable en una Java - AP.

El documento WO 03/021467 A divulga un sistema en el cual una aplicación java está asociada con una o más listas de permiso. No se establece ninguna reglamentación para modificar o actualizar la(s) lista(s) de actualizaciones después del suministro.

En consideración a dichos problemas, se ha realizado la presente invención para proporcionar un sistema que permita que las modificaciones de los derechos, los cuales han sido garantizados a los dispositivos de modificación, como por ejemplo las estaciones móviles, sean reflejados como restricciones de los comportamientos en los dispositivos de comunicaciones.

Divulgación de la invención

Para resolver el problema de la técnica convencional descrito con anterioridad, el objetivo de la presente invención consiste en proporcionar un procedimiento de comunicación de acuerdo con la reivindicación 1.

De acuerdo con la presente invención, se impide que un software que no debe ser ejecutado se ejecute en un dispositivo de comunicación, como por ejemplo una estación móvil, puesto que se comprueba si los datos de autorización, que indican el comportamiento., las cuales se permite que dirija una aplicación realizada por el software, son válidos o no mediante el acceso a un dispositivo externo; y, de acuerdo con un resultado de la comprobación, no se permite que se inicie el software no deseable.

Breve descripción de los dibujos

La Fig. 1 es un diagrama de bloques que muestra unas configuraciones de un sistema de comunicación de acuerdo con la presente invención.

La Fig. 2 es una figura que muestra un ejemplo del contenido del SDF utilizado en el sistema de comunicación.

La Fig. 3 e un diagrama de bloques que muestra unas configuraciones de una estación móvil, la cual es un componente del sistema de comunicación.

La Fig. 4 es una figura que muestra un ejemplo del contenido almacenado en una memoria no volátil de la estación móvil.

La Fig. 5 es un diagrama conceptual que muestra las estructuras funcionales de la estación móvil.

La Fig. 6 es un diagrama de flujo que muestra los procesos de la estación móvil para descargar e instalar el software de la Java-AP.

La Fig. 7 es un diagrama de flujo que muestra los procesos de la estación móvil para iniciar el software Java - AP.

La Fig. 8 es una figura que muestra un ejemplo de la respuesta de comprobación del SDF utilizado en el sistema de comunicación.

La Fig. 9 esa figura que muestra un ejemplo de respuesta de comprobación de las SDF utilizada en el sistema de comunicación.

La Fig. 10 es un gráfico de secuencia que muestra una serie de operaciones del sistema de comunicación.

Mejor modo de llevar a cabo la invención

A continuación, se analizará una forma de realización de la presente invención, con referencia a los dibujos. En los dibujos, los componentes comunes son designados mediante las mismas referencias numerales.

En primer lugar, se analizarán procedimientos que son seguidos por una estación móvil para descargar el software de la Java - AP dentro de un sistema de comunicación conocido. La estación móvil recibe en primer término un Archivo de Descriptores de Aplicaciones o un Fichero de Descripciones de Aplicaciones (designada en lo sucesivo como “ADF”) de un dispositivo servidor contenido en la World Wide Web (designado en lo sucesivo como “WWW”). A continuación, la estación móvil recibe un archivo Java Archive (designado en lo sucesivo como “Fichero JAR”).

Un ADF contiene unos datos dependientes de un correspondiente fichero JAR. Por ejemplo el ADF contiene unos datos que muestran una URL que indica una localización en la que el fichero JAR es almacenado (designado en lo sucesivo como “URL de paquetes”), unos datos que muestran un tamaño del fichero JAR, unos datos que muestran

el último tiempo de actualización del fichero JAR, y así sucesivamente. Estos datos son indispensables para el ADF. Después de que la estación móvil recibe el ADF, la estación móvil determina si un Java - AP que va a ser descargada, el cual está contenido en el fichero JAR, puede ser instalado en la estación móvil con referencia a los datos contenidos en el ADF, la comprobación de la capacidad de la memoria disponible, etc. Cuando la estación móvil determina que el Java - APP puede ser instalad, la estación móvil recibe el fichero JAR de un dispositivo servidor de la WWW utilizando una URL de paquetes contenida en el ADF. La estación móvil completa los procesos de descarga cuando recibe el fichero JAR que contiene el Java - APP. Después de la finalización de la descarga, el Java - APP descargada es instalada en la estación móvil y resulta posible la ejecución. Generalmente, los ADFs y los ficheros JAR son preparados por los CPs para proporcionar la Java - AP. El "software Java - AP" es un término para describir una combinación de un ADF y de un fichero JAR en el sistema de comunicación conocido.

Por otro lado cuando un Java - APP es descargado en una estación móvil en la presente forma de realización, en primer lugar un ADF correspondiente al Java - APP, la cual un usuario de la estación móvil desea descargar es suministrada a la estación móvil, entonces un Fichero de Descriptor de Seguridad o un Fichero de Descripción de Seguridad (designado en lo sucesivo como "SDF") el cual se corresponde con el Java - APP es suministrada a la estación móvil, y finalmente un fichero JAR el cual contiene el Java - APP, es suministrado a la estación móvil. En resumen, tres diferentes ficheros de ADF, SDF y JAR son suministrados a la estación móvil por este orden. El ADF y el fichero JAR son preparados por un CP que proporcional Java - APP de la misma forma que en el sistema de comunicación conocido. Sin embargo, el SDF está preparado por la portadora mencionada con anterioridad. De acuerdo con un contrato establecido entre la portadora y el CP. En la presente forma de realización, el "software Java - AP" se utiliza, así mismo, para describir una combinación de un ADF, un SDF y un fichero JAR.

El SDF es un fichero que describe el alcance de los derechos garantizados a un correspondiente Java - APP almacenada en una estación móvil, y una estación móvil ejecuta el Java - APP de acuerdo con el alcance de los derechos descritos en un SDF. Un servidor existente en la red de comunicación almacena los datos que indican si cada SDF está en un estado válido o en un estado no válido. Antes de que la estación móvil ejecuta el Java - APP el cual está instalado en la estación móvil, la estación móvil accede al servidor mencionado con anterioridad para verificar si el SDF correspondiente al Java - APP es válido o no es válido. En la exposición que sigue, este proceso de verificación será denominado "comprobación SDF". Si un resultado de la comprobación SDF está "en estado válido", la estación móvil ejecuta el Java - APP de acuerdo con el alcance de los derechos descritos en el SDF. Por otro lado, si el resultado de la comprobación SDF está en "estado no válido" la estación móvil no es autorizada a ejecutar el Java - APP.

De acuerdo con lo analizado anteriormente, dado que un estado de validez con respecto a cada SDF puede ser modificado en el servidor existente en la red de comunicación, la portadora puede fácilmente impedir que el Java - APP perjudicial sea ejecutada en estaciones móviles únicamente mediante el ajuste del SDF correspondiente al Java - APP como en el estado no válido.

La "Java - APP perjudicial" significa un Java - APP que puede dar instrucciones a una estación móvil para ejecutar una operación no esperada por un usuario de la estación móvil, como por ejemplo la lectura de datos secretos almacenados en una memoria de la estación móvil sin el consentimiento del usuario y la transmisión de los datos a un dispositivo externo. Cuando la portadora encuentra cualquier Java - APP perjudicial, la portadora solo tiene que fijar el estado del SDF del Java - APP perjudicial como un estado no válido. Cuando un contrato suscrito entre un CP y la portadora se convierte en válido, por ejemplo, debido a que ha expirado un periodo del contrato o a que el CP no ha pagado la tarifa indicada en el contrato dentro del plazo de vencimiento, la portadora puede, así mismo, hacer que el Java - APP no pueda ejecutar precisamente ajustando un estado del SDF del Java - APP como un estado no válido. Así mismo, en el caso de que el Java - APP se proporcione para un uso de prueba solo durante un periodo específico, la portadora puede hacer que el Java - APP no pueda ser ejecutada de la misma forma a la analizada con anterioridad, cuando el periodo de uso de prueba ha transcurrido.

Por el contrario, un estado de un SDF el cual se fija como un estado no válido puede ser modificado en un estado válido cuando la portadora quiere permitir que sea ejecutada un Java - APP correspondiente al SDF.

En la exposición siguiente, para describir una situación en la que una CPU ejecuta un Java - APP para proporcionar un grupo de funciones, a saber una Java - AP, se utilizará la expresión "se lleva a cabo una Java - AP".

(1) Configuración

A continuación, se analizarán unas configuraciones del sistema de comunicación.

Tal y como se muestra en la Fig. 1, el sistema de comunicación comprende un dispositivo servidor de CP 13 conectado a Internet 11, una red de comunicación de paquetes móviles 15, la cual se utiliza por la portadora para proporcionar servicios de comunicación de paquetes móviles hacia las estaciones móviles. Una estación móvil 16, la cual puede intercambiar paquetes de datos de forma inalámbrica con la red de comunicación de paquetes móviles 15, para comunicar con un dispositivo externo por medio de una red de comunicación de paquetes móviles 15, un dispositivo servidor de pasarela 17, el cual interconecta Internet 11 y la red de comunicación de paquetes móviles 15, y un dispositivo servidor de gestión 18 el cual está conectado a un dispositivo servidor de pasarela 17 por medio

de unas líneas dedicadas. Prácticamente, el sistema de comunicación puede comprender varias estaciones móviles y varios dispositivo servidor de CP, pero solo una de las estaciones móviles 16 y uno de los dispositivo servidor de CP 13 se muestran en la Fig. 1 para simplificar la figura.

A continuación, se analizarán detalles de cada componente del sistema de comunicación.

(1-1) Dispositivo servidor de CP

El dispositivo servidor de CP 13 presenta unos componentes de hardware y unos elementos característicos y unos dispositivo servidor de WWW general. Una memoria no volátil 13A del dispositivo servidor de CP 13 almacena varios ficheros JAR, cada uno de los cuales contiene un grupo de programas escritos en lenguaje de programación Java, a saber un Java - APP, y varios ADFs, cada uno de los cuales contiene datos con respecto a un correspondiente fichero JAR.

En el sistema de comunicación algunos Java - APPs son autorizados, cuyos correspondientes SDFs no existen. El comportamiento de una Java - AP, el cual es llevado a cabo por un Java - APP cuyo SDF existe se restringe sobre la base de su correspondiente SDF. Dicho Java - APP es llamado, en esta memoria descriptiva, “Java - APP de confianza” dado que su fiabilidad está garantizada por la portadora de acuerdo con el contrato suscrito entre la portadora y un CP el cual proporcional Java - APP. De modo similar, en la presente memoria descriptiva una Java - AP, la cual se lleva a cabo mediante un Java - APP de confianza se denomina una “Java - AP de confianza”. Un Java - APP cuyo ADF no existe es un Java - APP usual utilizado en un sistema de comunicación conocido y se llama, en la presente memoria descriptiva “Java - APP no de confianza” dado que su fiabilidad no está garantizada. De la misma forma, una Java - AP la cual se lleva a cabo mediante un Java - APP no de confianza se llama en la presente memoria descriptiva una Java - AP no de confianza. El dispositivo servidor de CP 13 puede almacenar tanto ADFs para Java - APPs de confianza como ADFs para Java - APPs no de confianza.

Un ADF para un Java - APP de confianza contiene algunos datos que están contenidos en un ADF para un Java - APP no de confianza, como por ejemplo una URL de paquetes que indiquen una localización de almacenamiento de un fichero JAR en la WWW, datos que indiquen el tamaño del fichero JAR como unos datos que indiquen el último tiempo de actualización del fichero JAR, etc. Un ADF para un Java - APP de confianza contiene, así mismo, unos datos llamados “APID” para identificar de modo exclusivo el fichero JAR paral Java - APP de confianza, y una URL llamado “SDF - URL” para indicar una localización de almacenamiento de un correspondiente SDF en la WWW. La estación móvil 16 obtiene un ADF correspondiente a un Java - APP de confianza del dispositivo servidor de CP 13, y obtiene un correspondiente SDF utilizando el SDF - URL contenido en el ADF.

En el análisis que sigue de la presente forma de realización, tal y como se indicó con anterioridad “software Java - AP” puede significar una combinación de un ADF, un SDF y un fichero JAR cuando el Java - APP contenido en el fichero JAR es un Java - APP de confianza, o una combinación de un ADF y un fichero JAR cuando el Java - APP es un Java - APP no de confianza.

(1-2) Dispositivo servidor de pasarela

El dispositivo servidor de pasarela 17 es gestionado por la portadora mencionada con anterioridad, y presenta unos componentes de hardware y unos elementos característicos y un dispositivo servidor de pasarela general, el cual interconecta una red de comunicación de paquetes móviles e Internet. El dispositivo servidor de pasarela 17 retransmite los datos mutuamente comunicados entre la red de comunicación de paquetes móviles 15 e Internet 11.

(1-3) Dispositivo servidor de gestión

El dispositivo servidor de gestión 18 es gestionado por la portadora indicada con anterioridad, e incorpora los componentes de hardware y los elementos característicos de un dispositivo servidor de WWW general. La memoria no volátil 18A del dispositivo de gestión 18 almacena varios SDFs, correspondiendo cada uno a un Java - APP de confianza. Cada SDF contiene, de acuerdo con lo expuesto con anterioridad, los datos que indican un alcance de los derechos que son garantizados al Java - APP de confianza. Cada SDF comprende así mismo los datos que indican si el SDF es válido o inválido. Antes de que la estación móvil 16 ejecute un Java - APP de confianza, la estación móvil 16 necesita acceder al dispositivo de gestión 18 y verificar si el SDF es válido o inválido. Sobre la base de un resultado de la verificación, la estación móvil 16 determina ejecutar o no el Java - APP de confianza.

La Fig. 2 ilustra un ejemplo de un mensaje de HTTP que es transmitido cuando los contenidos de un SDF son suministrados por el dispositivo servidor de gestión 18 a la estación móvil 16.

La línea comienza con el “Contenido - Tipo” mostrado en la Fig. 2, el cual es un campo que indica un cuerpo de entidad del HTTP. En este caso el campo contiene un parámetro de “application / x - sdf” que indica que el cuerpo de identidad se refiere a un SDF.

El parámetro “Sts” tiene por finalidad indicar si el SDF es válido o inválido y el llamado “estado SDF” en la explicación posterior. Cuando el estado SDF es “00”, significa que el SDF es válido, y cuando el estado SDF es “10” significa que el SDF es inválido. Por ejemplo, en el supuesto mostrado en la Fig. 2, dado que el parámetro “Sts” es

"00", el SDF es válido. El parámetro "Package URL" es la misma URL de paquetes contenida en un ADF tal como se expuso con anterioridad. El parámetro "CheckCnt" sirve para indicar un número de veces hasta el cual una estación móvil es autorizada a ejecutar el Java - APP en sucesión sin llevar a cabo una verificación del SDF, y el número de veces se llama "recuento para verificar el SDF" en la exposición posterior. Desde un punto de vista diferente, un recuento para la verificación del SDF son unos datos de frecuencia que muestran la frecuencia a la que una estación móvil es requerida para llevar a cabo las verificaciones del SDF.

En al presente forma de realización, por ejemplo, un recuento para la verificación del SDF es uno de los números enteros entre 1 y 999. En el ejemplo mostrado en la Fig. 2, el parámetro "CheckCnt" es "005", y significa que la estación móvil 16 es autorizada a ejecutar el Java - APP de confianza de manera sucesiva hasta 5 veces sin llevar a cabo una verificación del SDF.

El parámetro "CheqInt" sirve para indicar un intervalo de tiempo entre temporizaciones cuando las verificaciones del SDF deben llevarse a cabo, y se denomina "intervalo para la verificación del SDF" en un análisis posterior. Más concretamente, un intervalo para la verificación del SDF significa el número de días durante que la estación móvil 16 está autorizada para ejecutar el Java - APP de confianza sin llevar a cabo una verificación del SDF después de la última verificación del SDF. En la presente forma de realización, un intervalo para la verificación dl SDF es uno de números enteros entre 1 y 999. En el ejemplo mostrado en la Fig. 2, el parámetro "CheckInt" es "020" y significa que la estación móvil 16 es autorizada para ejecutar el Java - APP de confianza sin ejecutar una verificación del SDF durante 20 días una vez que la estación móvil 16 lleva a cabo una verificación del SDF, pero cuando pasan 20 días después de la última verificación del SDF, la estación móvil 16 es requerida para que lleve a cabo una verificación del SDF antes de ejecutar el Java - APP de confianza.

El parámetro "SuspendendCnt" sirve para indicar un número de veces hasta el cual una estación móvil es autorizada para ejecutar el Java - APP de confianza de forma sucesiva en un caso en el que la estación móvil no puede llevar a cabo una verificación del SDF, y el número de veces se denomina "recuento expandido para una verificación del SDF" en la exposición posterior.

La estación móvil 16 accede al dispositivo de servidor de gestión 18 de forma inalámbrica y lleva a cabo las verificaciones del SDF. De acuerdo con ello, puede resultar imposible para la estación móvil 16 llevar a cabo una verificación del SDF cuando, por ejemplo, la estación móvil 16 está fuera del área de servicio de la red de comunicación de paquetes de datos 15, es decir cuando no puede ser alcanzada, o cuando se produce un fallo de comunicación el cual puede producirse más frecuentemente en comunicaciones inalámbricas que en comunicaciones por cable. Por consiguiente, si no existe un parámetro como el de "SuspendedCnt", puede no ser posible que un usuario utilice un Java - APP de confianza cuando el usuario desea utilizarlo debido a un fallo de comunicaciones temporal, etc., lo cual es tremendamente incómodo para el usuario. Para evitar dicha incomodidad en la presente forma de realización, se autoriza que una estación móvil ejecute un Java - APP de confianza durante un número limitado de veces de forma sucesiva sin llevar a cabo una verificación del SDF incluso cuando la estación móvil no es capaz de llevar a cabo una verificación del SDF cuando no puede ser alcanzada o por fallo de las comunicaciones. En la presente forma de realización, un recuento expandido para la verificación del SDF es uno de los números enteros entre 1 y 999. En el ejemplo mostrado en la Fig. 2, el parámetro "SuspendedCnt" es "005", y significa que la estación móvil 16 es autorizada para ejecutar el Java - APP de confianza sin llevar a cabo una verificación del SDF hasta cinco veces seguidas incluso si no es capaz de llevar a cabo una verificación del SDF.

El parámetro "Lmd" sirve para indicar un tiempo y una fecha cuando el SDF fue actualizado por última vez en el dispositivo servidor de gestión 18, y el tiempo y la fecha se denominan "tiempo de la última actualización" en la exposición posterior.

En el ejemplo mostrado en la Fig. 2, el parámetro "Lmd" es "20020614120552", y significa que el SDF fue actualizado por última vez 5 minutos y 52 segundos más allá de las 12 en punto el 14 de Junio del año 2002. Los últimos tiempos de actualización son utilizados por el dispositivo servidor de gestión 18 para determinar si es necesario reflejar los contenidos actualizados de los SDFs en los datos de la estación móvil 16.

El parámetro "GetPrivateInfo", "UserMail", "MessageApp", "SetPhoneTheme", "SetLaunchTime", y "AllowedHost", sirven para indicar un alcance de los derechos que son garantizados al Java - APP de confianza ejecutado en la estación móvil 16, y son denominados "datos de permiso" en la exposición posterior.

Por ejemplo, los parámetros "GetPrivateInfo" sirve para indicar si un Java - APP de confianza realizado por el Java - APP de confianza es autorizado a utilizar una Interfaz de Programación de Aplicación de Confianza, designada en la sucesivo como "API de confianza", la cual es exigida de forma indispensable por la Java - AP cuando necesita referirse a los datos confidenciales almacenados en la estación móvil como por ejemplo los datos del directorio telefónico y los correos electrónicos no leídos. Cuando la Java - AP de confianza es autorizada a leer datos confidenciales, el parámetro "GetPrivateInfo" se establece como "Yes". De modo similar los parámetros "UserMail", "MessaeApp", "SetPhoneTheme", y "SetLaunchTime" sirven para indicar si la Java - AP de confianza está autorizada a llamar a las APIs de confianza, cada una de las cuales se corresponde con el parámetro respectivo, y cuando es autorizado, el parámetro se establece como "Yes". En el ejemplo mostrado en la Fig. 2, todos los parámetros se

setablecen como “Yes”, y ello significa que la Java - AP de confianza es autorizada a llamar a las correspondientes APIs de confianza.

El parámetro “AllowedHost” se dispone para indicar las URLs de los dispositivos de comunicación a las que una estación móvil está autorizado a acceder mientras que el Java - APP de confianza es ejecutado en la estación móvil, y las URLs son denominadas “URLs de acceso permitido” en la exposición posterior.

Un Java - APP que es descargado en la estación móvil 16 es generalmente ejecutado siguiendo un modelo de seguridad que se denomina en general un modelo de cajón de arena De acuerdo con el modelo de cajón de arena aunque la estación móvil 16 esté ejecutando un Java - APP, la estación móvil 16 está autorizada para comunicar solo con un servidor del que fue descargado el Java - APP. Con arreglo a dicha restricción estricta, es difícil proporcionar diversos tipos de programas de aplicación a un usuario de la estación móvil 16. Por consiguiente, cuando se ejecuta un Java - APP de confianza, a la estación móvil 16 se le permite comunicar con determinados dispositivos de comunicación externos que se especifican de antemano, además del servidor del que fue descargado el Java - APP de confianza. El parámetro “AllowedHost” muestra las URLs de los dispositivos de comunicación externos con los cuales se permite comunicar a una estación móvil. En el ejemplo mostrado en la Fig. 2, el parámetro muestra que a una estación móvil se le permite comunicar con los dispositivos de comunicación externos cuyas URLs contengan “http:// aaa.co.jp” o “htpp:// bbb.co.jp” utilizando el puerto de “8080”.

Cuando el parámetro de acceso de las URLs permitidas se establece como “any”, ello significa que a una estación móvil se le permite comunicar con cualquier dispositivo externo mientras el Java - APP de confianza es ejecutado. Dado que puede resultar difícil mantener la seguridad de los datos almacenados en la estación móvil si a la estación móvil se le permite comunicar con cualquier dispositivo de comunicación externo de acuerdo con el parámetro “any”, es, así mismo, posible especificar las URLs de determinados dispositivos de comunicación externos con los cuales a una estación móvil no se le permite comunicar. El parámetro “DisallowedHost” está preparado para este fin, y las URLs de los dispositivos de comunicación externos con los cuales no se permite comunicar a una estación móvil, que son denominados “URLs de acceso prohibido” en la exposición posterior, se ajustan al parámetro.

(1-4) Estación móvil

La estación móvil 16 comprende, tal y como se ilustra en la Fig. 3, el software del Sistema Operativo (designado en la sucesivo como “OS”), la ROM 16A la cual almacena un programa de entorno Java para construir un entorno para elaborar Java - APPs operables y otros diversos programas de aplicación nativa, una UCP 16B que ejecuta los programas almacenados en la ROM 16A, una unidad de pantalla 16C, una memoria no volátil 16D, una RAM 16E, una unidad de comunicación 16F, una unidad operativa 16G y un temporizador 16H. Estos componentes están conectados entre sí por medio de un bus de datos.

La unidad de pantalla 16C comprende, por ejemplo, un panel de una pantalla de cristal líquido y un circuito excitador del panel, y muestra las imágenes de acuerdo con los datos suministrados por la UCP 16B. La unidad de comunicación 16F comprende una antena y una unidad de comunicación inalámbrica, que comunica de forma inalámbrica paquetes de datos con la red de comunicación de paquetes móviles 15, y retransmite paquetes de datos entre la UCP 16B y la red de comunicación de paquetes móviles 15. La unidad de comunicación 16F comprende un CÓDEC, un micrófono y un altavoz para las comunicaciones de voz, y hace posible que la estación móvil 16 dirija comunicaciones de voz a través de una red telefónica móvil (no mostrada) la cual incorpora un sistema de conmutación de líneas. La unidad operativa 16G comprende un hardware, como por ejemplo un teclado para las operaciones de un usuario, y proporciona a la UPC 16B ciertas señales de acuerdo con las operaciones llevadas a cabo por el usuario. El temporizador 16H mantiene el año, mes, fecha y tiempo actual (designado en lo sucesivo precisamente como “tiempo presente”). Para conseguir que el temporizador 16H mantenga el tiempo actual correcto, la estación móvil 16 puede recibir de forma regular datos que indiquen el tiempo actual preciso utilizando un canal de control a partir de una estación de base (no mostrada) de la red de comunicación de paquetes móviles 15, y ajustar el tiempo mantenido por el temporizador 16H.

La memoria no volátil 16D es, por ejemplo, una Memoria de Acceso Aleatorio Estática (SRAM) o una Memoria de Solo Lectura Programable y Eléctricamente Borrable (EEPROM). La memoria no volátil 16D almacena el software Java - AP el cual es descargado de los servidores comprendidos en la WWW.

A continuación se analizarán detalles de un ADF y de un SDF de un software Java - AP de confianza, el cual está almacenado en una memoria no volátil 16D.

La Fig. 4 muestra un ejemplo de unos datos que son generados de acuerdo con un ADF y con el SDF mostrado en la Fig. 2 del software Java - AP de confianza, y almacenados en la memoria no volátil 16D. Tal y como se muestra en la Fig. 4, los datos almacenados en la memoria no volátil 16D contienen un campo de datos de “número de veces de ejecución”, un campo de datos de “numero de días pasados” y un campo de datos de “recuento expandido utilizado para la verificación del SDF” los cuales se corresponden con el “recuento para la verificación del SDF”, un “intervalo para la verificación del SDF” y un recuento expandido para la verificación del SDF, respectivamente.

Una pluralidad de episodios de ejecución indica un número de veces que la estación móvil 16 ejecutó el Java - APP de confianza después de la ejecución más reciente de una verificación del SDF. Después de que la pluralidad de

tiempos de ejecución alcanza el recuento para la verificación del SDF, a saber 5 en el supuesto ejemplar mostrado en la Fig. 4, la estación móvil 16 determina que se requiere llevar a cabo una verificación del SDF antes de que ejecute el Java - APP de confianza la próxima vez.

Una pluralidad de días pasados indica un periodo de tiempo, a saber, una pluralidad de días que pasaron después de la ejecución más reciente de una verificación del SDF por la estación móvil 16. Los días pasados son calculados sobre la base del tiempo guardado por el temporizador 16H de manera continua. Después de que la pluralidad de días pasados alcanza el intervalo para la verificación del SDF, a saber 20 en el supuesto ejemplar mostrado en la Fig. 4, la estación móvil 16 determina que debe llevarse a cabo una verificación del SDF antes de que ejecute la vez siguiente el Java - APP de confianza. Un recuento expandido utilizado para la verificación del SDF significa una pluralidad de veces que la estación móvil 16 ejecutó el Java - APP de confianza sin llevar a cabo la verificación del SDF cuando una verificación del SDF se requiere dado que la estación móvil 16 no fue capaz de llevar a cabo una verificación del SDF. Después de que el recuento expandido utilizado para la verificación del SDF alcanza el recuento expandido para la verificación del SDF, a saber, 5 en el supuesto mostrado en la Fig. 4, la estación móvil 16 no ejecuta el Java - APP de confianza a menos que lleve a cabo con éxito una verificación del SDF.

Cuando una fuente de alimentación (no mostrada) de una estación móvil 16 es activada, la UPC 16B lee el programa del OS almacenado en la ROM 16A y ejecuta el programa del OS utilizando la RAM 16E como área de trabajo. Siguiendo las instrucciones del programa del OS, la UPC 16B establece un entorno de OS en la estación móvil, tal y como se ilustra en la Fig. 5. Después de que se ha establecido el entorno de OS, la UPC 16B es habilitada para identificar las instrucciones proporcionadas por el usuario sobre la base de las señales suministradas por la unidad operativa 16G, y para ejecutar el procesamiento de datos requerido de acuerdo con las instrucciones.

En lo sucesivo, cuando se utilice una descripción como “el OS ejecuta una operación “, ello significa que la UCP 16B ejecuta la operación siguiendo las instrucciones proporcionadas por el programa OS. De la misma forma, cuando se utiliza una descripción como “la aplicación ejecuta una operación”, significa que la UPC 16B ejecuta el programa de aplicación y ejecuta la operación de acuerdo con las instrucciones suministradas por el programa de aplicación. Diversas aplicaciones mostradas en la Fig. 5, como por ejemplo la Java - AP, el Gestor de Aplicaciones Java (designado en lo sucesivo como “JAM”) y el AP de directorio telefónico son ejemplos de dicha aplicación.

Cuando el usuario da instrucciones a la estación móvil para que ejecute un programa de comunicación, el cual es uno de los programas de aplicación nativa, el OS inicia el programa de comunicación y lleva a cabo una AP de comunicación en la estación móvil 16. Una vez que se ha realizado el AP de comunicación, el usuario es capaz de efectuar una llamada de voz a un dispositivo de comunicaciones externo. Cuando el usuario da instrucciones a las estaciones para ejecutar un programa de directorio telefónico, el cual es, así mismo, uno de los programas de aplicación nativa, el OS inicia el programa de directorio telefónico y realiza un programa de directorio telefónico en la estación móvil 16. El usuario es capaz de hacer posible que la estación móvil 16 muestre en pantalla, utilice y edite los datos del directorio telefónico (designados en lo sucesivo como “datos de directorio telefónico” almacenados en la memoria no volátil 16D mediante la utilización del AP de directorio telefónico. Cuando el usuario da instrucciones a la estación móvil 16 para que ejecute un programa de exploración de la Web, el cual es, así mismo, uno de los programas de aplicación nativa, el OS inicia el programa de exploración de la Web y realiza una exploración de la Web en la estación móvil 16.

Cuando el usuario da instrucciones a la estación móvil 16 para que ejecute un programa de JAM, el cual es, así mismo uno de los programas de aplicación nativa, el OS inicia el programa de JAM y realiza un programa JAM en la estación móvil 16. El JAM muestra al usuario una lista de Java - APPs que están instalados en la estación móvil 16, y ejecuta uno de los Java - APPs que es seleccionado por el usuario. Más concretamente, si la instrucción dictada por el usuario es una solicitud para iniciar uno de los Java - APPs, el primer programa de entorno de Java - AP es ejecutado para establecer un entorno de Java - AP en la estación móvil 16, a continuación el Java - APP seleccionado por el usuario es ejecutado para realizar la Java - AP en el entorno Java - AP. Un entorno Java - AP contiene, por ejemplo, una Máquina Virtual K (KVM) la cual es una máquina virtual Java de peso ligero sintonizada para un dispositivo de comunicación móvil, como por ejemplo la estación móvil 16, y las APIs las cuales ofrecen diversas funciones a las Java - APs. Las APIs, las cuales están preparadas para las Java - APs están en las categorías de APIs de confianza, las cuales solo se permite que utilicen las Java - APs, es decir las Java - APs las cuales son realizadas por los Java - APPs de confianza y las Java - APs no de confianza, las cuales se permite que utilicen todas las Java - APs.

El JAM proporciona, así mismo, las funciones de gestión del comportamiento de los Java - APs.

Por ejemplo, cuando el JAM recibe una solicitud para ejecutar un Java - APP, el JAM determina si se requiere una verificación del SDF con respecto al SDF del Java - APP, y si el JAM determina que se requiere la verificación del SDF, lleva a cabo la verificación del SDF. Para la finalidad indicada, el JAM proporciona unas funciones de generación y actualización de datos que indican la pluralidad de momentos de ejecución, la pluralidad de días pasados, y el recuento expandido utilizado para la verificación del SDF, los cuales van a ser almacenados en la memoria no volátil 16D. Así mismo, cuando un resultado de la verificación del SDF indica que el SDF es válido, el JAM permite que el Java - APP sea ejecutado y, después de que la Java - AP se ha realizado, el JAM restringe el comportamiento de la Java - AP de acuerdo con los datos de permiso contenidos en el SDF. Por otro lado, cuando el

resultado de la verificación del SDF indica que el SDF es inválido, el JAM no permite que el Java - APP sea ejecutado. Cuando el JAM determina, sobre la base del resultado de la verificación del SDF, que es necesario utilizar el SDF, el JAM de nuevo accede al dispositivo servidor de gestión 18, obtiene el SDF actualizado, y actualiza los datos almacenados en la estación móvil 16.

(2) Funcionamiento

A continuación se analizarán las operaciones de sistema de comunicación, cuyas configuraciones se describieron en las líneas anteriores.

(2-1) Instalación del Java - APP

Cuando un JAM recibe una solicitud para descargar un Java - APP del explorador de la Web, el JAM inicia una serie de procesos para instalar el Java - APP en la estación móvil 16. La Fig. 6 ilustra un flujo de los procesos.

Tal y como se ilustra en la Fig. 6, el JAM, en primer término, descarga del dispositivo servidor de CP 13 un ADF correspondiente al Java - APP que va a ser descargado (etapa S11). Más concretamente, el JAM genera una solicitud del HTTP que contiene un ADF - URL, transmite la solicitud del HTTP al dispositivo servidor de CP 13 y recibe una respuesta del HTTP que contiene el ADF procedente del dispositivo servidor de CP 13 en respuesta a la solicitud del HTTP. En este proceso, un ADF - URL contenido en una solicitud del HTTP puede ser una entrada realizada manualmente por el usuario, o seleccionada por el usuario entre las URLs candidatos embebidas en una página web descrita en el Lenguaje de Composición de Hipertexto (en adelante designado como “HTML”). El JAM da instrucciones a la memoria no volátil 16D para que almacene los datos contenidos en el ADF descargado, como por ejemplo una APID, una URL de paquetes y un SDF - URL y un ADF - URL, tal y como se muestra en la Fig. 4.

A continuación, el JAM determina si es posible que el Java - APP sea descargado e instalado en la estación móvil 16 sobre la base de los datos del ADF almacenado en la memoria no volátil 16D (etapa S12). El JAM puede llevar a cabo la determinación de la misma forma que en el sistema conocido, por ejemplo mediante la comparación del tamaño de un fichero JAR indicado en el ADF y el espacio libre disponible de la memoria no volátil 16D donde el fichero JAR va a ser almacenado.

Si el JAM determina que la instalación es posible (etapa S12; Sí), el JAM determina si el Java - APP que va a ser descargado es un Java - APP de confianza o un Java - APP no de confianza (etapa S13). Más concretamente, el JAM verifica si una APID y un SDF - URL están contenidos en los datos almacenados en la etapa S11 y si están contenidos el JAM determina que hay un SDF que corresponde al Java - APP, lo que significa que el Java - APP es un Java - APP de confianza. Por el contrario si una APID y un SDF - URL no están contenidos el JAM determina que el Java - APP es un Java - APP no de confianza. Ahora se supone que una APID y un SDF - URL están contenidos en el ADF descargado, y el JAM determina que el Java - APP que va a ser descargado es un Java - APP de confianza (etapa S13; Sí).

El JAM recibe un SDF correspondiente al Java - APP de confianza del dispositivo servidor de gestión 18 (etapa S14). Más concretamente, el JAM establece una conexión de TCP entre el dispositivo servidor de gestión 18, genera una solicitud del HTTP que contiene el SDF - URL obtenido del ADF, transmite la solicitud del HTTP al dispositivo servidor de gestión 18, recibe una respuesta del HTTP (por favor, remítase a la Fig. 2) en respuesta a la solicitud del HTTP, y desconecta la conexión del HTTP).

El JAM verifica si el SDF recibido contiene los datos apropiados (etapa S15). Más concretamente, el JAM verifica si el SDF recibido es descrito de acuerdo con un determinado formato, verifica si una APID contenida en el SDF y la APID almacenada en la memoria no volátil 16D coinciden, etc., y si todos los resultados de las verificaciones son afirmativos, el JAM determina que el SDF recibido contiene los datos apropiados correctos (etapa S15; Si). A continuación, el JAM da instrucciones a la memoria no volátil 16D para almacenar los datos contenidos en el SDF tal y como se muestra en la Fig. 4.

El JAM descarga el fichero JAR extrayéndolo del dispositivo servidor del CP 13 (etapa S16). Más concretamente, el JAM genera una solicitud del HTTP que contiene la URL de paquetes almacenados en la memoria no volátil 16D, transmite la solicitud del HTTP al dispositivo servidor de CP 13 y recibe una respuesta del HTTP que contiene el fichero JAR desde el dispositivo servidor CP 13 en respuesta a la solicitud del HTTP.

El JAM da instrucciones al archivo JAR recibido para almacenar la memoria no volátil 16D e instalal Java - APP de confianza contenido en el fichero JAR siguiendo los procedimientos normales para la instalación de un Java - APP (etapa S17). A continuación, el JAM notifica al usuario que la instalación se ha completado de manera satisfactoria (etapa S18).

Después de que la instalación se ha completado, cuando el Java - APP de confianza es ejecutado, el JAM supervisa el comportamiento de la Java - AP de confianza la cual es realizada por el Java - APP, y restringe la Java - AP de confianza a la utilización solo de ciertas APIs de confianza de acuerdo con los datos de permiso contenidos en el SDF y almacenados en la memoria no volátil 16D.

Durante los procesos mencionados con anterioridad, cuando se determina que el Java - APP no es capaz de ser instalado (etapa S12; No), o cuando el SDF no contiene los datos apropiados (etapa S15; No), el JAM notifica al usuario que la instalación falló (etapa S20) y restaura las configuraciones software en la estación móvil 16 a un estado anterior a la etapa S11.

Durante los procesos mencionados con anterioridad, cuando se determina que el Java - APP que va a ser instalado es un Java - APP no de confianza (etapa S13; No), el JAM descarga el fichero JAR del dispositivo servidor de CP 13 de acuerdo con una URL de paquetes contenida en el ADF siguiendo un procedimiento ordinario (etapa S19), e instala un Java - APP contenido en el fichero JAR de acuerdo con los procedimientos normales para la instalación de un Java - APP (etapa S17). A continuación, el JAM notifica al usuario que la instalación se ha completado de manera satisfactoria (etapa S18).

(2-2) Inicio del Java - APP

A continuación se analizarán las operaciones para el inicio de un Java - APP, con referencia a la Fig. 7.

Cuando el JAM recibe del usuario una solicitud para iniciar un Java - APP, el JAM determina si el Java - APP que va a ser iniciado es un Java - APP de confianza o un Java - APP no de confianza (etapa S31). Cuando se determina que va a ser iniciado un Java - APP de confianza (etapa S31; Sí), el JAM determina si se requiere una verificación del SDF (S32). Mas concretamente, el JAM verifica los datos almacenados en la memoria no volátil 16D y determina que se requiere una verificación del SDF cuando el número de episodios de ejecución es mayor que el recuento para la verificación del SDF, o cuando el número de días pasados es mayor que el intervalo para la verificación del SDF. Por otro lado, el JAM determina que no se requiere una verificación del SDF cuando ambas condiciones no concurren.

Por ejemplo, de acuerdo con los datos mostrados en la Fig. 4, las dos condiciones mencionadas con anterioridad no concurren, el JAM determina que no se requiere una verificación del SDF (etapa S32; No). A continuación el JAM añade 1 al número de los tiempos de ejecución y actualiza el número almacenado en la memoria no volátil 16D (etapa S33), e inicial Java - APP de confianza (etapa S34).

Después de que el mismo Java - APP de confianza es iniciado reiteradamente por el JAM, de acuerdo con lo analizado con anterioridad, el número de los episodios de ejecución puede resultar el mismo que el recuento para la verificación del SDF. En el supuesto, el JAM determina que se requiere una verificación del SDF (etapa S32; Si), genera una solicitud del HTTP para solicitar unas operaciones de verificación del SDF, y transmite la solicitud del HTTP al dispositivo servidor de gestión 18 (etapa S35). La solicitud del HTTP contiene una APID y unos datos que indican el último episodio de actualización del SDF correspondiente al Java - APP de confianza que va a ser iniciado, en concreto el valor del parámetro “Lmd”.

Después de la recepción de la solicitud del HTTP, el dispositivo servidor de gestión 18 extrae la APID de la solicitud del HTTP, lee un SDF que contiene la APID almacenada en la memoria no volátil 18A y verifica un estado de validez del SDF. El dispositivo servidor de gestión 18 extrae, así mismo, el último episodio de actualización de la solicitud del HTTP, compara el último episodio de actualización extraído y un último episodio de actualización contenido en el SDF leído a partir de la memoria no volátil 18A, y determina si es necesario actualizar el SDF almacenado en la estación móvil 16. Más concretamente, el dispositivo servidor de gestión 18 determina que la actualización es necesaria si el último episodio de actualización extraído de la solicitud del HTTP es más antiguo que el último episodio de actualización existente en el SDF de la memoria no volátil 18A, y determina que la actualización no es necesaria si ambos últimos episodios de actualización son lo mismos. De acuerdo con un resultado de la determinación mencionada con anterioridad, el dispositivo servidor de gestión 18 genera una respuesta del HTTP que contiene los datos que indican un estado de validez del SDF, y transmite la respuesta del HTTP a la estación móvil 18 como una respuesta de verificación del SDF.

La Fig. 8 y la Fig. 9 muestran ejemplos de la respuesta de verificación del SDF.

La Fig. 8 muestra una respuesta de verificación del SDF que es generada cuando el SDF es válido. En el caso, tal y como se muestra en la Fig. 8, el valor del parámetro “Sts”, el cual indica el estado de validez del SDF, se establece como “00” que significa “válido”. Por otro lado, la Fig. 9 muestra una respuesta de verificación del SDF la cual es generada cuando el SDF es inválido. En el caso, tal y como se muestra en la Fig. 9, el valor del parámetro “Sts” se fija en “10” significando “inválido”. Cuando el SDF es válido pero está actualizado, el valor del parámetro “Sts” se establece como “01” que indica que es necesario que el SDF sea actualizado (no mostrado en los dibujos). Cuando un SDF que contiene la APID recibida de la estación móvil 16 no está almacenado en la memoria no volátil 18A, el valor del parámetro “Sts” se fija en “99” que indica que el SDF podría no ser encontrado (no se muestra en los dibujos).

Cuando la estación móvil 16 recibe la respuesta de verificación del SDF (etapa S36; Sí), el JAM verifica el contenido de la respuesta de verificación del SDF (etapa S37).

Por ejemplo, cuando el JAM recibe una respuesta de verificación del SDF, tal y como se muestra en la Fig. 8, el JAM determina que el SDF es válido (etapa S38; válido), borra el numero de los episodios de ejecución, el número de los

días pasados y el recuento expandido utilizado para la verificación del SDF almacenado en la memoria no volátil 16D (etapa S38), e inicial Java - APP de confianza (etapa S34).

Por otro lado, cuando el JAM recibe una respuesta de verificación del SDF, tal y como se muestra en la Fig. 9, el JAM determina que el SDF es inválido (etapa S37; inválido), muestra un mensaje de que el Java - APP no es capaz de ser iniciado, y restaura las configuraciones software en la estación móvil 16 a un estado previo a la etapa S31 (etapa S39).

Cuando la respuesta de verificación del SDF recibida indica que el SDF está actualizado, el JAM determina que el SDF almacenado en la memoria no volátil 16D es requerido para que sea actualizado (etapa S37; actualizar) y lleva a cabo unos procedimientos para actualizar el SDF (etapa S40). Más concretamente, de modo similar a las operaciones de la etapa S14 de la Fig. 6, el JAM recibe los datos del SDF actualizado desde el dispositivo servidor de gestión 18, verifica que los datos recibidos del SDF son apropiados, y almacena los datos del SDF actualizado en la memoria no volátil 16D.

Durante las operaciones mencionadas con anterioridad, si el JAM no es capaz de iniciar o no consigue completar una verificación del SDF debido a que, por ejemplo, la estación móvil 16 está en un estado inalcanzable o se produce un fallo de la comunicación (etapa S36; No), el JAM verifica los datos almacenados en la memoria no volátil 16D y determina si el recuento expandido utilizado para la verificación del SDF es inferior al recuento expendido para la verificación del SDF (etapa S41).

Cuando el recuento expandido utilizado para la verificación del SDF es inferior al recuento expandido de la verificación SDF (etapa S41; Sí), el JAM añade 1 al recuento expandido utilizado para la verificación del SDF y actualiza el recuento almacenado en la memoria no volátil 16D (etapa S42). A continuación, el JAM inicial Java - APP de confianza (etapa S34).

Por otro lado, cuando el recuento expandido utilizado para la verificación del SDF es el mismo o mayor que el recuento expandido para la verificación del SDF (etapa S41; No), el JAM muestra un mensaje de que no es capaz de llevar a cabo una verificación del SDF y el Java - APP no puede ser iniciado, y restaura las configuraciones software en la estación móvil 16 a un estado previo a la etapa S31 (etapa S39).

(2-3) Ejemplo de operaciones del sistema de comunicación global para hacer operativo un Java - APP de confianza

A continuación se analizará, con referencia a la Fig. 10, un ejemplo de una serie de operaciones para hacer operativo un Java - APP de confianza, el cual sigue las etapas mencionadas con anterioridad.

En el análisis siguiente, las operaciones que son llevadas a cabo mediante el JAM u otras aplicaciones, como por ejemplo las Java - APs, son descritas como operaciones que son llevadas a cabo por la “estación móvil 16”.

En la Fig. 10, cuando la estación móvil 16 recibe del usuario una solicitud para descargar un Java - APP, la estación móvil 16 genera una solicitud del HTTP m1 la cual contiene un ADF - URL de un ADF correspondiente al Java - APP que va a ser descargado y transmite la solicitud del HTTP m1 al dispositivo servidor de CP 13.

En respuesta a la solicitud del HTTP m1, el dispositivo servidor de CP 13 genera la respuesta del HTTP m2 la cual contiene el correspondiente ADF y transmite la respuesta del HTTP m2 a la estación móvil 16.

Después de la recepción de la respuesta del HTTP m2 , la estación móvil 16 almacena los datos del ADF en la memoria no volátil 16D. Si la estación móvil 16 determina que es posible instalar el Java - APP en la estación móvil 16, la estación móvil 16 genera la solicitud del HTTP m3 la cual contiene un SDF - URL de un SDF, y transmite la solicitud del HTTP m3 al dispositivo servidor de gestión 18.

Después de la recepción de la solicitud del HTTP m3, el dispositivo servidor de gestión 18 genera una respuesta del HTTP m4 la cual contiene el SDF, y transmite la respuesta del HTTP m4 a la estación móvil 16 en respuesta a la solicitud del HTTP m3.

Después de la recepción del HTTP m4 la estación móvil 16 verifica que los contenidos del SDF son apropiados, y almacena el SDF en la memoria no volátil 16D. A continuación, la estación móvil 16 genera una solicitud del HTTP m5 la cual contiene una URL de paquetes, y transmite la solicitud del HTTP m5 al dispositivo servidor de CP 13.

En respuesta a la solicitud del HTTP m5 el dispositivo servidor de CP 13 genera la respuesta del HTTP m6 la cual contiene un fichero JAR, y transmite la respuesta del HTTP m6 a la estación móvil 16. El fichero JAR contiene el Java - APP de confianza que va a ser descargado.

Después de la recepción de la respuesta del HTTP m6, la estación móvil 16 almacena el fichero JAR recibido en la memoria no volátil 16D, y lleva a cabo los procedimientos para la instalación del Java - APP de confianza.

Después de que se ha completado la instalación del Java - APP de confianza, cuando la estación móvil 16 recibe una instrucción para iniciar el Java - APP, la estación móvil 16 determina si se requiere una verificación del SDF o

no, y si se determina que se requiere una verificación del SDF, la estación móvil 16 transmite la solicitud del HTTP m7 la cual contiene una APID y un último episodio de actualización del SDF al dispositivo servidor de gestión 18 como una solicitud de verificación del SDF.

Después de la recepción de la solicitud del HTTP m7, el dispositivo servidor de gestión 18 genera una respuesta del HTTP m8 la cual contiene los datos que indican un estado de validez del SDF correspondiente a la APID contenida en la solicitud del HTTP m7, y transmite la respuesta del HTTP m8 a la estación móvil 16 como respuesta de verificación del SDF.

Después de la recepción de la respuesta del HTTP m8, la estación móvil 16 verifica el contenido de la respuesta de verificación del SDF. Por ejemplo, cuando la respuesta de la verificación del SDF indica que es necesario que el SDF sea actualizado, la estación móvil 16 actualiza el SDF. En concreto, la estación móvil 16 genera una solicitud del HTTP m9 la cual contiene el SDF - URL , y transmite la solicitud del HTTP m9 al dispositivo servidor de gestión 18.

Después de la recepción de la solicitud del HTTP m9, el dispositivo servidor de gestión 18 genera una respuesta del HTTP m10 la cual contiene el SDF identificado por el SDF - URL contenido en la solicitud del HTTP m9 y transmite la respuesta del HTTP m10 a la estación móvil 16.

Después de la recepción de la respuesta del HTTP m10, la estación móvil 16 inicial Java - APP y ejecuta el Java - APP de acuerdo con el contenido del SDF actualizado.

(2-4) Gestión del comportamiento de la estación móvil 16 mientras el Java - APP es ejecutado

A continuación, se analizará la gestión del comportamiento, de la estación móvil 16 cuando se ejecuta el Java - APP mencionado con anterioridad.

(2-4-1) Java - APP no de confianza

Después de que el Java - APP de no confianza está instalado en la estación móvil 16 de acuerdo con los procedimientos de instalación mencionados con anterioridad, se inicial Java - APP no de confianza, lo que da como resultado la realización de una Java - AP no de confianza en la estación móvil 16. A continuación, la siguiente gestión del comportamiento es dirigida por la estación móvil 16.

Cuando una API cuya utilización es solicitada por lA Java - AP no de confianza es una API no de confianza, la estación móvil 16 permite que la Java - AP no de confianza utilice la API no de confianza, dado que, tal y como expuso con anterioridad, permite que la API no de confianza sea utilizada por otro tipo de Java - AP. En resumen, la Java - AP no de confianza es autorizada a llamar a la API no de confianza.

Cuando una API de la que se solicita que sea utilizada por la Java - AP no de confianza es una API de confianza, la estación móvil 16 verifica si un SDF correspondiente a la Java - AP está almacenado en la memoria no volátil 16D. Dado que ningún SDF correspondiente a la Java - AP no de confianza está almacenado en la memoria no volátil 16D, la estación móvil 16 prohíbe que la Java - AP no de confianza utilice la API de confianza. En resumen, el Java - AP no de confianza no es autorizada a llamar a la API de confianza.

(2-4-2 Java - AP de confianza

Después de que un Java - APP de confianza está instalado en la estación móvil 16, el Java - APP de confianza es iniciado, y se realiza una Java - AP de confianza en la estación móvil 16, la gestión del comportamiento siguiente es dirigida por la estación móvil 16.

Cuando una API cuya utilización se solicita por la Java - AP de confianza, es una API no de confianza, por supuesto la estación móvil 16 permite que la Java - AP de confianza utilice la API no de confianza, de acuerdo con lo expuesto con anterioridad. En resumen, la Java - AP de confianza es autorizada a llamar a la API no de confianza.

Cuando una API cuya utilización se solicita por la Java - AP de confianza es una API no de confianza, dado que un SDF correspondiente a la Java - AP de confianza está almacenado en la memoria no volátil 16D, la estación móvil 16 puede permitir que la Java - AP de confianza utilice la API de confianza. Sin embargo, en este supuesto, el comportamiento de la Java - AP de confianza está restringido de acuerdo con los datos de permiso contenidos en el SDF.

Por ejemplo, si el parámetro “GetPrivateInfo” de los datos de permiso se establece como “Yes”, la Java - AP de confianza es autorizado a leer los datos del directorio telefónico y los datos que contienen los correos electrónicos no leídos existentes en la memoria no volátil 16D. Al mismo tiempo, la Java - AP de confianza es autorizada a comunicar solo con los dispositivos externos cuyas URLs están contenidas en el parámetro “AllowedHost” existente en los datos de permiso.

De acuerdo con lo analizado con anterioridad, la estación móvil 16 permite el comportamiento de la Java - AP de confianza que cumple las condiciones indicadas en los datos de permisos contenidos en el SDF correspondiente a la

Java - AP de confianza, y prohíbe el comportamiento que no cumpla con las condiciones. De acuerdo con ello, el usuario puede utilizar no solo las Java - APs del sistema conocido, a saber, las Java - APs no de confianza, sino también las Java - APs de confianza que son autorizados a comportarse de una manera más flexible que los Java - APs no de confianza, disfrutando de esta manera de un mayor grado de comodidad.

En el sistema de comunicación de acuerdo con esta forma de realización, la estación móvil 16 recibe un ADF, un SDF, y un archivo JAR, de forma separada y por este orden, lo cual conlleva determinados efectos de acuerdo con lo que sigue.

Tal y como se analizó con anterioridad, el software Java - AP, cada uno de los cuales comprende un ADF y un fichero JAR, generalmente son diseñados y realizados por los CPs, y cada CP ofrece, en su propio sitio (dispositivo servidor de CPs 13 en la Fig. 1) en Internet para suministrar sus software Java - AP a cualquier usuario. De acuerdo con ello, el usuario generalmente decide descargar o no descargar el software Java - AP después de acceder a los sitios de los CPs y leer las explicaciones introductorias de sus software Java - AP mostrados en las páginas de los sitios. Cuando el usuario decide descargar determinado software Java - AP el usuario necesita operar la estación móvil de él / ella para que le dé instrucciones para llevar a cabo los procedimientos de descarga. Con dicho fin, las páginas de los sitios mencionados con anterioridad, generalmente contienen las URLs de los ficheros que van a ser descargados junto con los marcadores de anclaje. Desde el punto de vista de los CPs, es la forma más fácil para embeber las URLs de los ADFs en sus páginas, dado que los ADFs son gestionados por los CPs y las URLs de los ADFs son bien conocidos por los CPs. Por otro lado, para embeber las URLs de los SDFs en las páginas de los CPs, los CPs tienen que obtener las URLs de la portadora y mantener la actualización de las URLs. Sin embargo, si un ADF, un SDF y un fichero JAR son distribuidos hasta una estación móvil por separado en este orden, no hay necesidad de que un CP obtenga, embeba y actualice una URL de un SDF, lo cual es bastante útil.

Así mismo, el procedimiento mencionado con anterioridad de distribución de tres clases de ficheros en un cierto orden es apropiado cuando se tienen en cuenta los procedimientos de actualización del software Java - AP dentro del sistema de comunicación conocido. En el sistema de comunicación conocido actualmente utilizado, cuando un usuario solicita la actualización del software Java - AP, la estación móvil de él / ella primero verifica los datos contenidos en un ADF correspondiente al software Java - AP e intenta obtener la versión actualizada del fichero JAR utilizando una URL de paquetes contenida en el ADF. En concreto, una estación móvil verifica en primer término un ADF, a continuación descarga un fichero JAR elegido como objetivo.

En comparación con los procedimientos para actualizar el software Java - AP actualmente adoptados, tal y como se ha analizado con anterioridad, los procedimientos para actualizar el software Java - AP adoptados en el sistema de comunicación de acuerdo con la presente forma de realización ofrecen un flujo similar. Es decir, en el sistema de comunicación de acuerdo con la presente forma de realización, una estación móvil verifica en primer término un ADF de la misma forma que en el sistema conocido, y después de obtener un SDF - URL contenido en el ADF, la estación móvil tiene un fichero JAR utilizando el SDF - URL de manera similar a la del sistema conocido. De acuerdo con ello, es posible introducir un sistema de comunicación de acuerdo con la presente forma de realización sin efectuar cambios significativos en el sistema que se está actualmente utilizando. En comparación con el procedimiento propuesto con anterioridad, si un SDF, un ADF, y un fichero JAR son distribuidos a una estación móvil por este orden, cuando el software Java - AP es actualizado, una vez que la estación móvil obtiene un ADF, la estación móvil puede obtener una versión actualizada del fichero JAR la próxima vez refiriéndose al ADF obtenido anteriormente, es decir, sin referirse al SDF más reciente. Dado que el SDF puede ser modificado cuando la versión actualizada del fichero JAR es descargada en la estación móvil, la versión actualizada del fichero JAR sin referencia con el SDF más reciente, puede provocar problemas de seguridad. Por consiguiente, un ADF, un SDF, y un fichero JAR deben ser distribuidos a una estación móvil por este orden.

(3) Modificación

La presente invención no está limitada a la forma de realización analizada con anterioridad y puede ser modificada dentro del alcance técnico de la presente invención. A continuación se formulan ejemplos de dichas modificaciones.

En el sistema de comunicación mencionado con anterioridad, el comportamiento de las Java - APs es gestionado con respecto a las APIs que son utilizadas por las Java - APs y los dispositivos externos con los cuales las Java - APs comunican, pero puede establecerse cualquier otra clase de condiciones sobre los usos para gestionar el comportamiento. “Recursos” pueden ser recursos software, como por ejemplo APIs recursos de red de comunicación, como por ejemplo un dispositivo de comunicación con el cual comunique la Java - AP y recursos hardware.

“Hardware” puede ser un hardware de una estación móvil como por ejemplo una memoria, un altavoz, un micrófono, un controlador de radiación de infrarrojos, y un DiodoFotoemisor (LED), y el “hardware” puede, así mismo, consistir en unos dispositivos externos que operen de forma conjunta con la estación móvil, como por ejemplo un módulo de Identidad de Usuario (UIM) y un módulo de Identidad de Abonados (SIM).

Por supuesto, los “recursos de la red de comunicación” no deben quedar limitados a los recursos existentes en la red de comunicación, como pueden ser los dispositivos de comunicaciones con los cuales la estación comunica. Tal

y como se indicó con anterioridad, la estación móvil comunica con la red de comunicaciones móviles de forma inalámbrica. Durante las comunicaciones inalámbricas, la estación móvil utiliza recursos de comunicaciones inalámbricas, como por ejemplo canales inalámbricos asignados por la red de comunicaciones móviles. Los recursos de comunicaciones inalámbricas pueden, así mismo, ser utilizados de acuerdo con el término "recursos de redes de comunicaciones" indicado con anterioridad. Al mismo tiempo, la estación móvil puede utilizar recursos de comunicaciones tales como trayectorias de transmisión para comunicaciones de paquetes de datos y trayectorias de comunicaciones de un sistema de conmutación de líneas para comunicaciones de voz sobre niveles de protocolos de comunicaciones los cuales estén situados más altos que el nivel de los protocolos de comunicaciones sobre los cuales son utilizados los recursos de comunicaciones inalámbricas. Los recursos de comunicaciones sobre niveles de protocolos de comunicaciones pueden, así mismo, ser utilizados como los "recursos de redes de comunicaciones" mencionados con anterioridad.

"Recursos software" pueden ser APIs, clases, paquetes, etc., de acuerdo con lo analizado con anterioridad. Los recursos software pueden proporcionar diversos tipos de funciones, y las funciones de procesamiento de datos, como por ejemplo los datos de encriptación y las funciones de comunicación de datos, como por ejemplo los datos de transmisión y recepción, las aplicaciones, como por ejemplo el explorador de la Web, son funciones típicas. En la presente invención, los recursos software compuestos por los dispositivos externos mencionados con anterioridad, pueden, así mismo ser utilizados como parámetros de condiciones para restringir el comportamiento de las Java - APs.

En la forma de realización mencionada con anterioridad, el JAM prohíbe determinados tipos de comportamiento de los Java - APs de confianza antes de que se comporten de acuerdo con los datos de permiso. Sin embargo, el JAM puede gestionar el comportamiento de las Java - APs mediante la terminación obligatoria de los procesos de las Java - APs de confianza cuando el JAM detecta que las Java - APs de confianza intentan comportarse de una manera que no cumple con las condiciones indicadas en los datos de permiso.

En el sistema de comunicación referido con anterioridad, el software es distribuido a las estaciones móviles, pero el software puede ser distribuido a otros tipos de dispositivos de comunicación.

En el sistema de comunicación referido con anterioridad, la portadora puede funcionar como un CP. En concreto, un dispositivo servidor de gestión y un dispositivo servidor de CP pueden estar situados en el mismo nodo de comunicación.

Los programas mencionados con anterioridad ejecutados por una UPC de una estación móvil, como por ejemplo un programa de JAM y un programa de OS, y sus datos relacionados, pueden ser suministrados cuando son almacenados en los medios de registro desde los cuales la UPC es capaz de leer los programas y los datos, por ejemplo medios de registro magnéticos, medios de registros ópticos, y una ROM. Los programas y los datos, pueden, así mismo, ser suministrados a las estaciones móviles por medio de redes de comunicaciones, como por ejemplo Internet.

REIVINDICACIONES

1.- Un procedimiento de comunicación entre:

• un sistema de comunicación que incluye (a) un programa de aplicación Java que proporciona un dispositivo servidor (18) que almacena un programa de aplicación Java para suministrar una serie de funciones, (b) un dispositivo servidor de gestión (18) que almacena unos datos de descriptores de seguridad SDF que contienen unos datos de permiso que indican un alcance de los derechos que van a ser utilizados por el programa de aplicación Java, (c) un dispositivo servidor (13) que proporciona unos datos de descriptores de aplicación Java ADF que almacena los datos de los descriptores de aplicación Java ADF que indican una localización del almacenamiento del programa de aplicación Java y una localización de almacenamiento de los datos de descriptores de seguridad SDF, y

• un dispositivo de comunicación (16) que ejecuta el programa de aplicación Java,

comprendiendo el procedimiento:

a) la transmisión desde el sistema de comunicación hasta el dispositivo de comunicación (16) de los datos de descriptores de aplicación Java ADF;

b) la transmisión de los datos (URL) que indican la localización del almacenamiento de los datos de descriptores de seguridad SDF contenidos en los datos de descriptores de descripción ADF desde el dispositivo de comunicación (16) hasta el sistema de comunicación;

(c) la transmisión de los datos de descriptores de seguridad SDF desde el sistema de comunicación hasta el dispositivo de comunicación (16) sobre la base de los datos que indican la localización del almacenamiento de los datos de descriptores de seguridad SDF;

d) el almacenamiento de los datos de descriptores de seguridad SDF en el dispositivo de comunicación (16);

e) la transmisión de los datos que indican la localización del almacenamiento del programa de aplicación Java contenidos en los datos de descriptores de seguridad SDF desde el dispositivo de comunicación (16) hasta el sistema de comunicación;

f) la transmisión del programa de aplicación Java desde el dispositivo de comunicación hasta el dispositivo de comunicación (16) sobre la base de los datos (URL) que indican la localización del almacenamiento del programa de aplicación Java;

g) la instalación, en el dispositivo de comunicación (16), del programa de aplicación Java transmitido desde el sistema de comunicación hasta el dispositivo de comunicación (16); en el que

h) el dispositivo de comunicación (16) accede al servidor de gestión existente en la red de comunicación y verifica, antes del inicio del programa de aplicación Java, si los datos de descriptores de seguridad SDF almacenados en el dispositivo de comunicación (16) son válidos; y

i) permite que el programa de aplicación Java sea ejecutado en el dispositivo de comunicación (16) solo cuando se determina que los datos de descriptores de seguridad SDF se consideran válidos sobre la base de un resultado de la verificación, en el que el programa de aplicación Java es ejecutado para proporcionar una función en línea con el alcance de los derechos indicado por los datos de permiso contenidos en los datos de descriptores de seguridad SDF almacenados en el dispositivo de comunicación (16).

FIG. 1

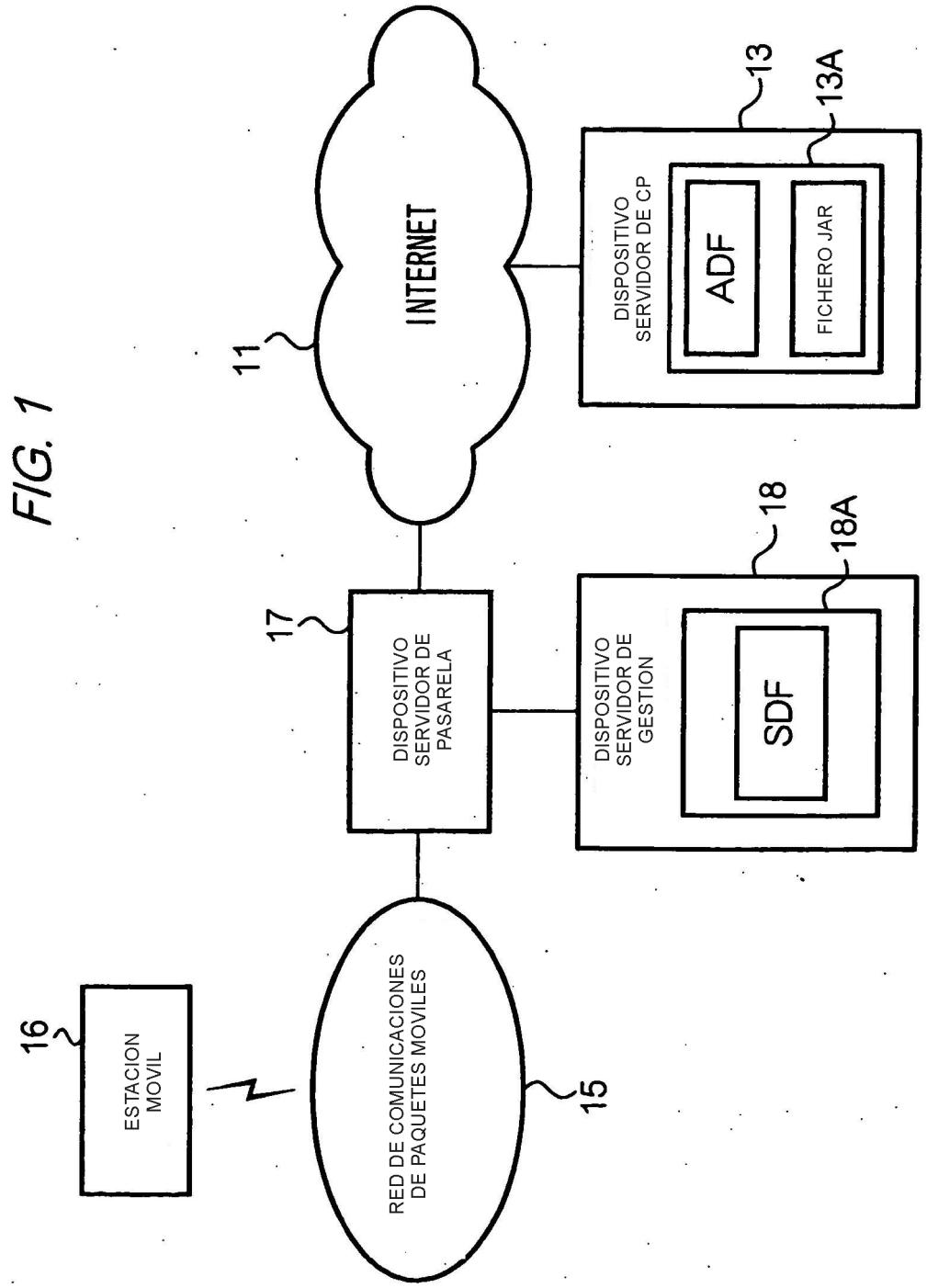

INTERNET
11
DISPOSITIVO SERVIDOR DE CP
13
13A
ADF
FICHERO JAR
DISPOSITIVO SERVIDOR DE PASARELA
17
DISPOSITIVO SERVIDOR DE GESTION
18
18A
SDF
RED DE COMUNICACIONES DE PAQUETES MOVILES
15
ESTACION MOVIL
16

FIG. 2

HTTP/1.0 200 OK
Content-Type:application/x-sdf
•
•
<CR><LF>
Sts = 00 <CR><LF>
PackageURL = http://cpserver.com:8080/Demo.JAR <CR><LF>
CheckCnt = 005 <CR><LF>
CheckInt = 020 <CR><LF>
SuspendedCnt = 005 <CR><LF>
Lmd=20020614120552 <CR><LF>
GetPrivateInfo = yes <CR><LF>
UseMailer = yes <CR><LF>
MessageApp = yes <CR><LF>
SetPhoneTheme = yes<CR><LF>
SetLaunchTime = yes <CR><LF>
AllowedHost = http://aaa.co.jp http://bbb.co.jp:8080 <CR><LF>

FIG. 5

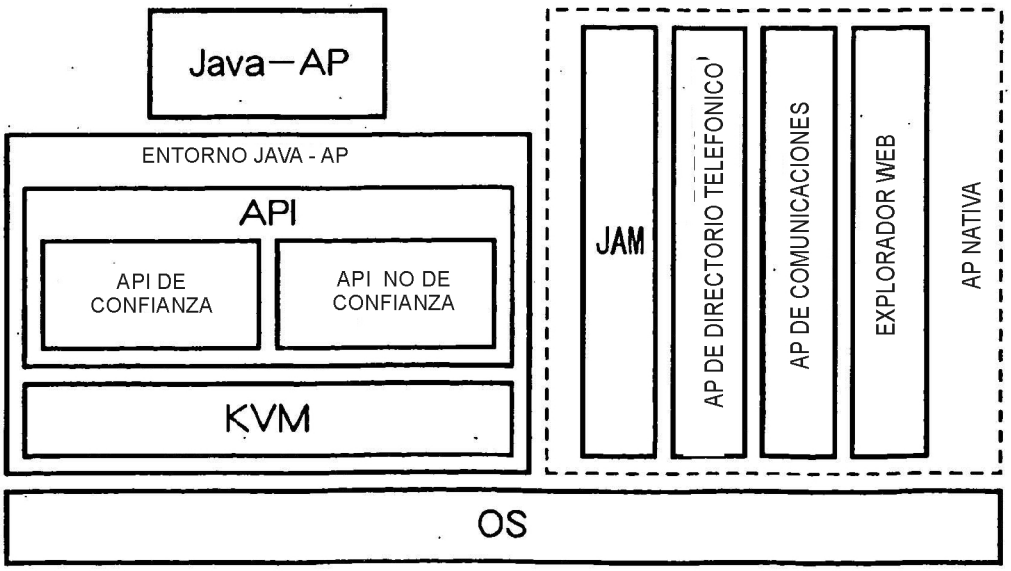


FIG. 3

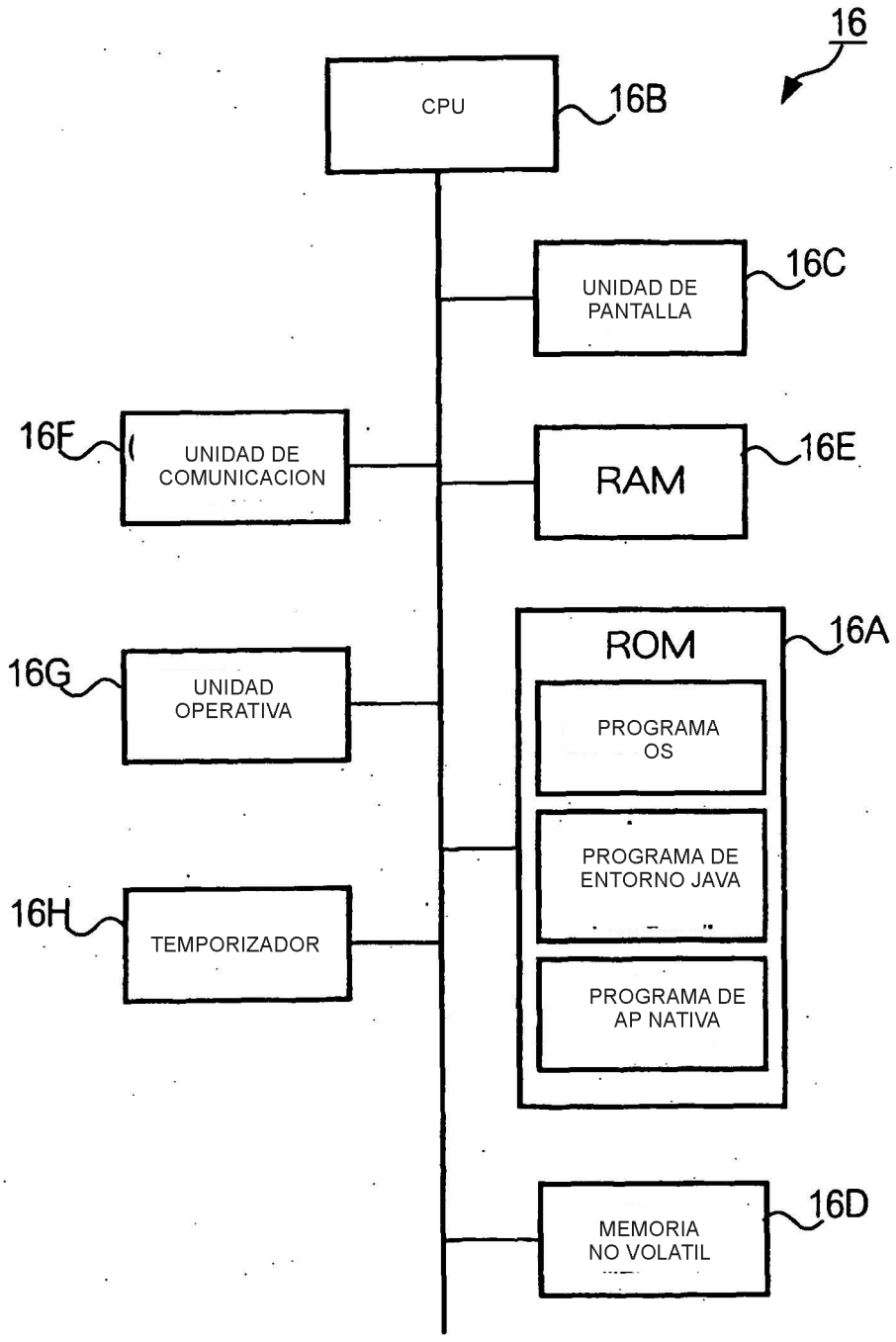
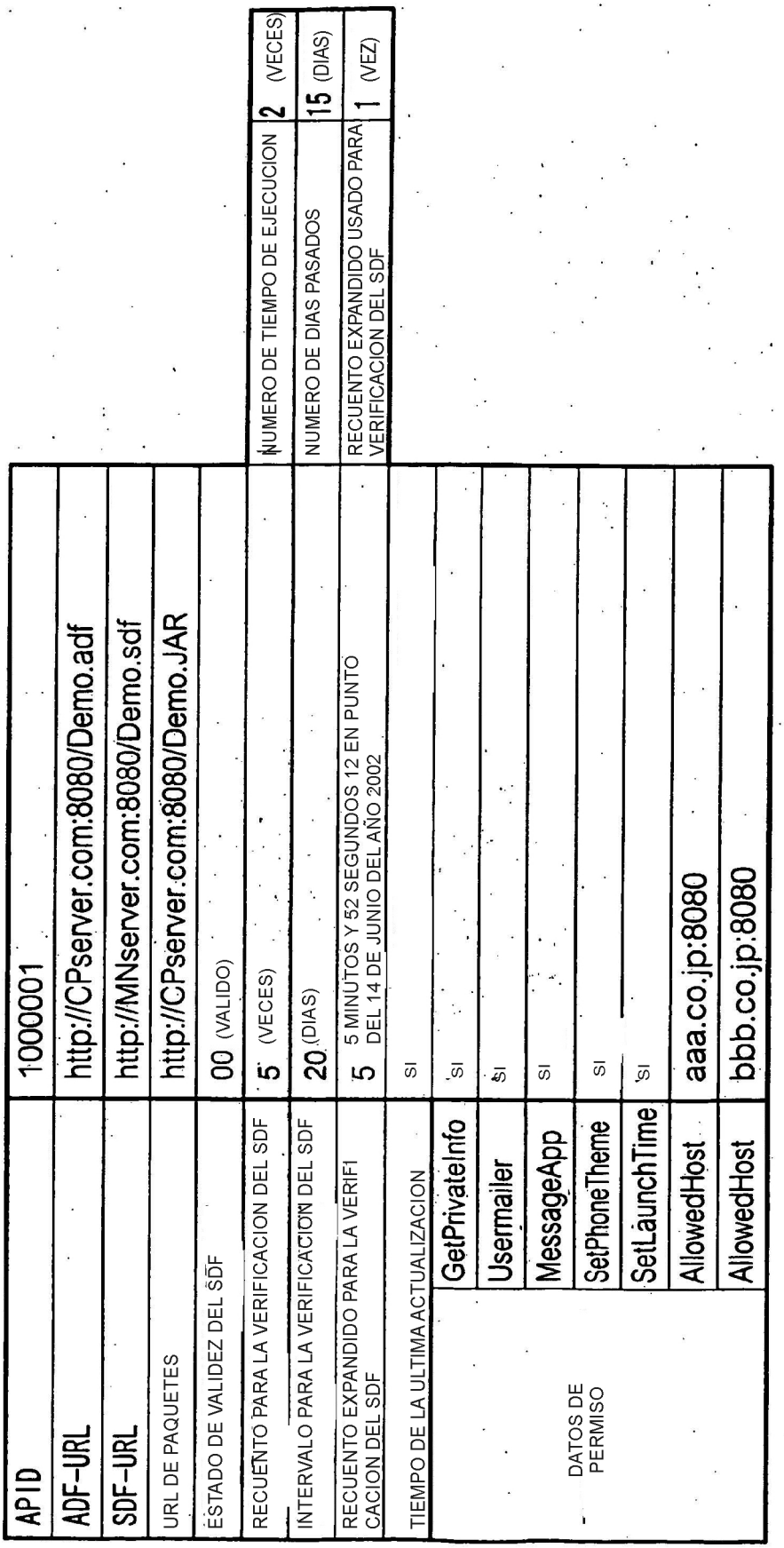

16
CPU
16B
UNIDAD DE PANTALLA
16C
16F
UNIDAD DE COMUNICACION
RAM
16E
ROM
16A
16G
UNIDAD OPERATIVA
PROGRAMA OS
PROGRAMA DE ENTORNO JAVA
16H
TEMPORIZADOR
PROGRAMA DE AP NATIVA
MEMORIA NO VOLATIL
16D

FIG. 4



APID	1000001		
ADF-URL	http://CPserver.com:8080/Demo.adf		
SDF-URL	http://MNserver.com:8080/Demo.sdf		
URL DE PAQUETES	http://CPserver.com:8080/Demo.JAR		
ESTADO DE VALIDEZ DEL SDF	00 (VALIDO)		
RECUENTO PARA LA VERIFICACION DEL SDF	5 (VECES)	NUMERO DE TIEMPO DE EJECUCION	2 (VECES)
INTERVALO PARA LA VERIFICACION DEL SDF	20 (DIAS)	NUMERO DE DIAS PASADOS	15 (DIAS)
RECUENTO EXPANDIDO PARA LA VERIFICACION DEL SDF	5 5 MINUTOS Y 52 SEGUNDOS 12 EN PUNTO DEL 14 DE JUNIO DEL AÑO 2002	RECUENTO EXPANDIDO USADO PARA VERIFICACION DEL SDF	1 (VEZ)
TIEMPO DE LA ULTIMA ACTUALIZACION	SI		
DATOS DE PERMISO	GetPrivateInfo	SI	
	Usermailer	SI	
	MessageApp	SI	
	SetPhoneTheme	SI	
	SetLaunchTime	SI	
	AllowedHost	aaa.co.jp:8080	
	AllowedHost	bbb.co.jp:8080	

FIG. 6

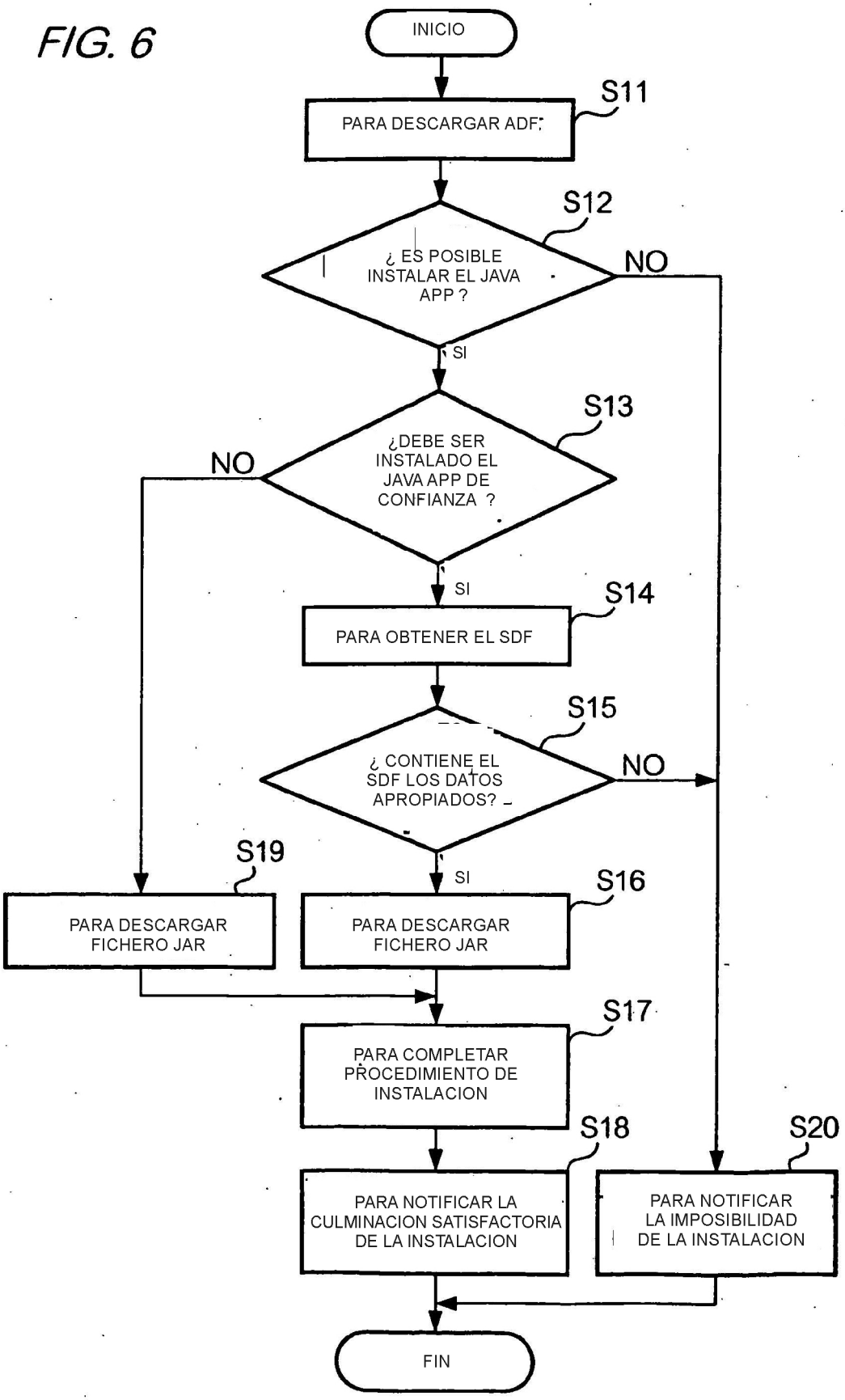

INICIO
S11
PARA DESCARGAR ADF
S12
¿ ES POSIBLE INSTALAR EL JAVA APP ?
NO
SI
S13
¿DEBE SER INSTALADO EL JAVA APP DE CONFIANZA ?
NO
SI
S14
PARA OBTENER EL SDF
S15
¿ CONTIENE EL SDF LOS DATOS APROPIADOS?
NO
SI
S19
PARA DESCARGAR FICHERO JAR
S16
PARA DESCARGAR FICHERO JAR
S17
PARA COMPLETAR PROCEDIMIENTO DE INSTALACION
S18
PARA NOTIFICAR LA CULMINACION SATISFACTORIA DE LA INSTALACION
S20
PARA NOTIFICAR LA IMPOSIBILIDAD DE LA INSTALACION
FIN

FIG. 7

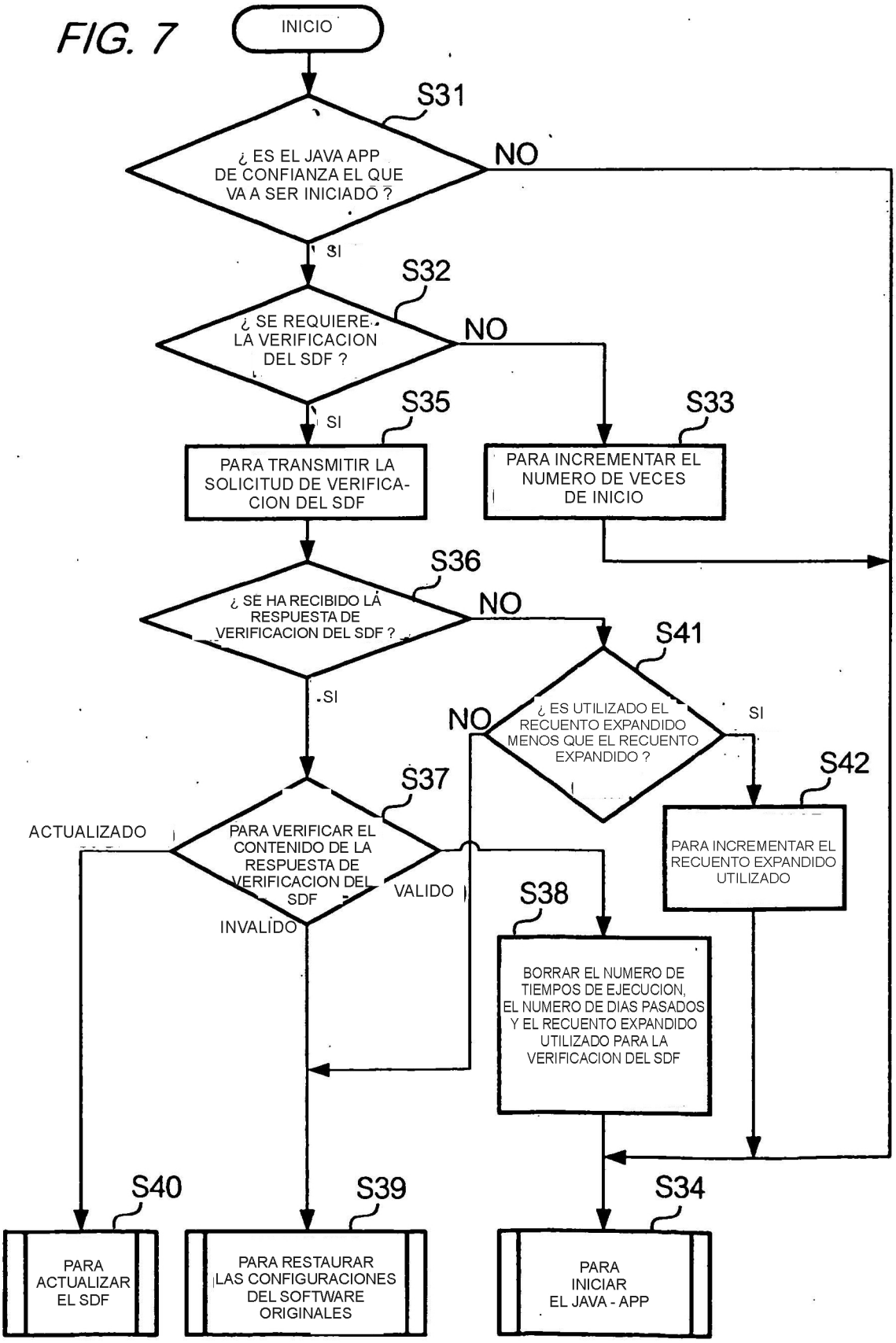

INICIO
S31
¿ ES EL JAVA APP DE CONFIANZA EL QUE VA A SER INICIADO ?
NO
SI
S32
¿ SE REQUIERE LA VERIFICACION DEL SDF ?
NO
SI
S35
PARA TRANSMITIR LA SOLICITUD DE VERIFICACION DEL SDF
S33
PARA INCREMENTAR EL NUMERO DE VECES DE INICIO
S36
¿ SE HA RECIBIDO LA RESPUESTA DE VERIFICACION DEL SDF ?
NO
SI
S41
¿ ES UTILIZADO EL RECUENTO EXPANDIDO MENOS QUE EL RECUENTO EXPANDIDO ?
NO
SI
S42
PARA INCREMENTAR EL RECUENTO EXPANDIDO UTILIZADO
S37
PARA VERIFICAR EL CONTENIDO DE LA RESPUESTA DE VERIFICACION DEL SDF
ACTUALIZADO
VALIDO
INVALIDO
S38
BORRAR EL NUMERO DE TIEMPOS DE EJECUCION, EL NUMERO DE DIAS PASADOS Y EL RECUENTO EXPANDIDO UTILIZADO PARA LA VERIFICACION DEL SDF
S40
PARA ACTUALIZAR EL SDF
S39
PARA RESTAURAR LAS CONFIGURACIONES DEL SOFTWARE ORIGINALES
S34
PARA INICIAR EL JAVA - APP

FIG. 8

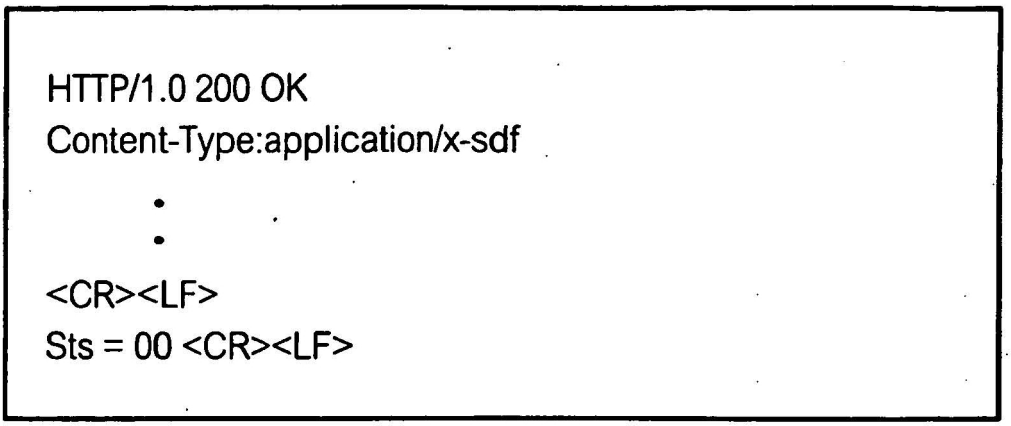

HTTP/1.0 200 OK
Content-Type:application/x-sdf
•
•
<CR><LF>
Sts = 00 <CR><LF>

FIG. 9

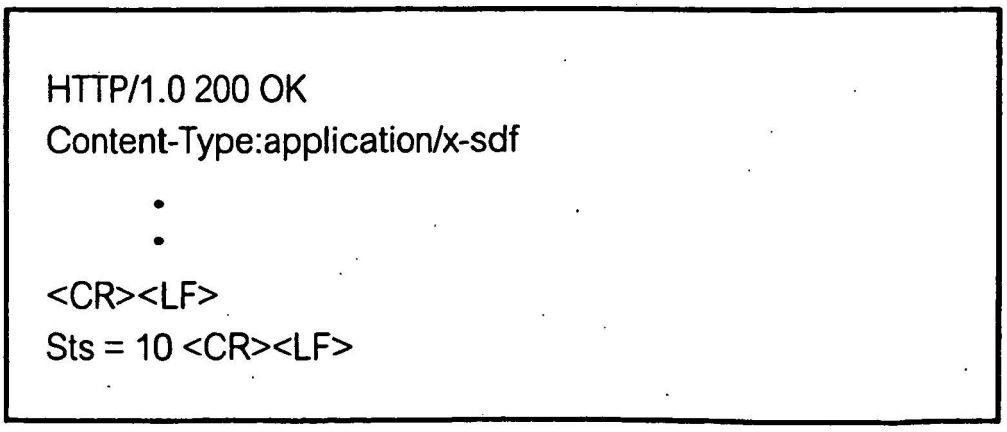
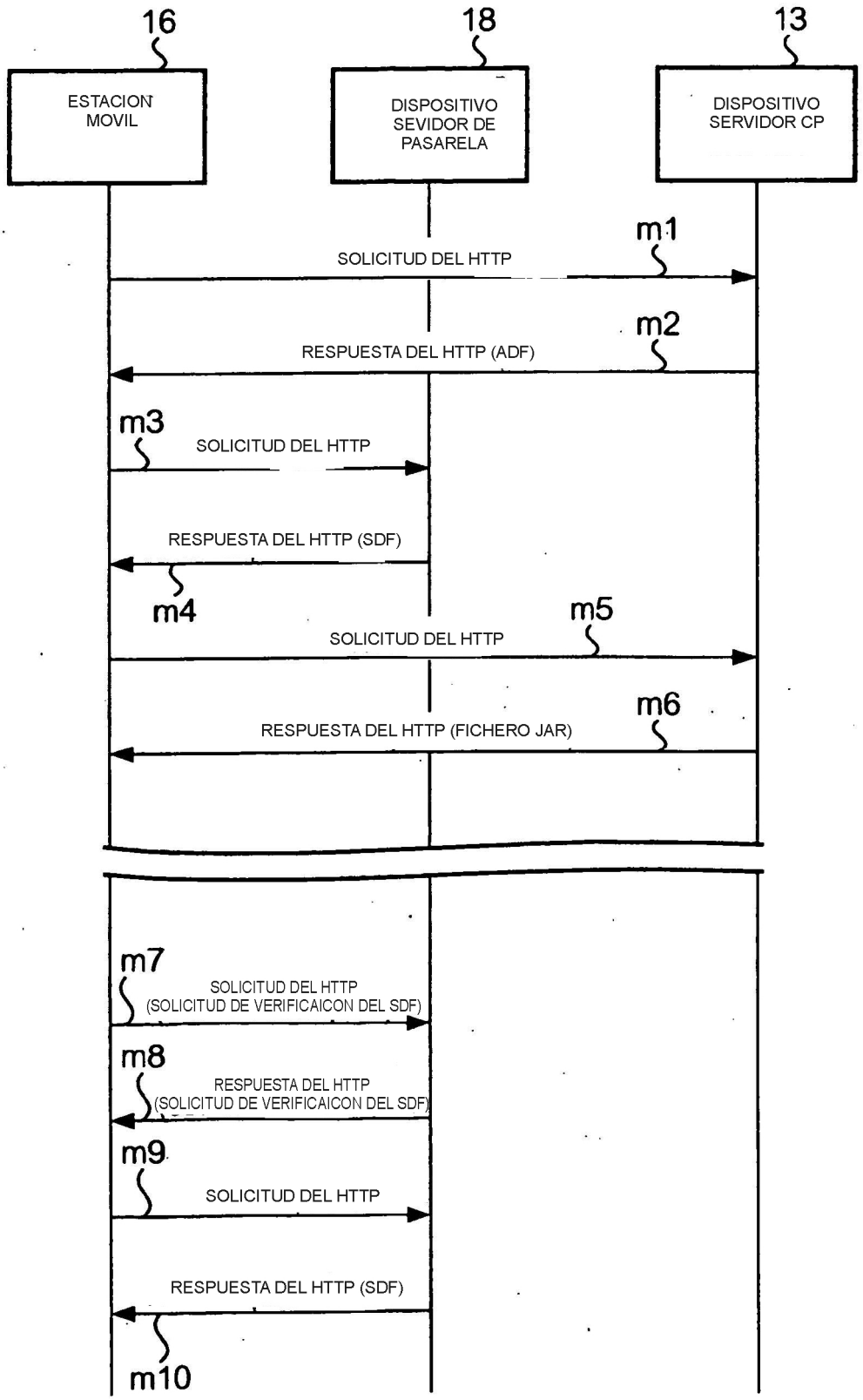

HTTP/1.0 200 OK
Content-Type:application/x-sdf
•
•
<CR><LF>
Sts = 10 <CR><LF>

FIG. 10


16
18
13
ESTACION MOVIL
DISPOSITIVO SEVIDOR DE PASARELA
DISPOSITIVO SERVIDOR CP
m1
SOLICITUD DEL HTTP
m2
RESPUESTA DEL HTTP (ADF)
m3
SOLICITUD DEL HTTP
RESPUESTA DEL HTTP (SDF)
m4
m5
SOLICITUD DEL HTTP
m6
RESPUESTA DEL HTTP (FICHERO JAR)
m7
SOLICITUD DEL HTTP
(SOLICITUD DE VERIFICAICON DEL SDF)
m8
RESPUESTA DEL HTTP
(SOLICITUD DE VERIFICAICON DEL SDF)
m9
SOLICITUD DEL HTTP
RESPUESTA DEL HTTP (SDF)
m10