(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 855 199**

(21) Número de solicitud: 202090021

(51) Int. Cl.:

H04W 4/00 (2008.01)
G06Q 30/00 (2013.01)

(12) PATENTE DE INVENCIÓN CON EXAMEN B2

(22) Fecha de presentación:
12.12.2017

(43) Fecha de publicación de la solicitud:
23.09.2021

Fecha de modificación de las reivindicaciones:
07.12.2023

Fecha de concesión:
08.11.2024

(45) Fecha de publicación de la concesión:
15.11.2024

(73) Titular/es:
MOXIBLE, S.L. (100.0%)
Avenida de Oporto 66, 2º
28019 Madrid (Madrid) ES

(72) Inventor/es:
PÉREZ GARCÍA, Isidoro

(74) Agente/Representante:
PÉREZ GARCÍA, Isidoro

(54) Título: **SISTEMA Y MÉTODO DE IDENTIFICACIÓN ÚNICA DE DISPOSITIVOS ELECTRÓNICOS**

(57) Resumen:
Sistema y método de identificación única de dispositivos electrónicos de usuario (102) que son capaces de ejecutar aplicaciones móviles que integran un SDK APP (104) y un navegador web que a su vez integra un SDK web que utiliza una serie de nodos de escucha y procesamiento de señales de radiofrecuencia emitidas por equipos de usuario para, mediante cruce de datos de análisis en tiempo real, generar identificadores únicos de usuario propagados entres los nodos por un nodo servidor.
Los identificadores se actualizan mediante nuevos datos (firmas) de forma incremental combinando varias firmas.

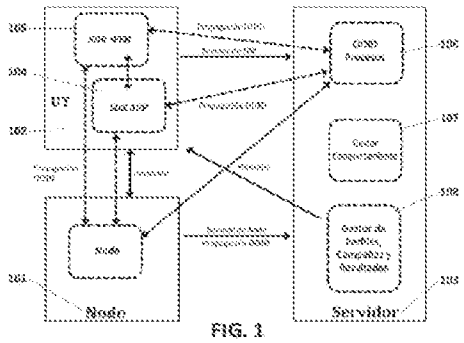

FIG. 1

DESCRIPCIÓN

SISTEMA Y MÉTODO DE IDENTIFICACIÓN ÚNICA DE DISPOSITIVOS ELECTRÓNICOS

OBJETO DE LA INVENCIÓN

El objeto de la invención se enmarca en el campo técnico de las tecnologías de la información y telecomunicaciones.

Más concretamente el objeto de la invención va dirigido a la gestión de asociación e identificación de dispositivos electrónicos tales como terminales en redes de comunicaciones.

ANTECEDENTES DE LA INVENCIÓN

Hoy en día, la mayor parte de personas llevamos todo el día con nosotros un terminal móvil personal (en adelante terminal de usuario, o terminal de usuario por sus siglas en inglés UT), que está presente en nuestra vida o entidad offline (cuando visitamos por diversos sitios físicos con el teléfono en el bolsillo encendido pero no activo) y también en nuestra vida o entidad online (cuando nos visitamos sitios online través páginas web o visitamos aplicaciones móviles a través de dicho terminal).

Por tanto, en el ámbito de esta solución consideramos que es el mismo usuario quién a través de un terminal de usuario utiliza una aplicación móvil, o utiliza/visita una página web desde este terminal, que visita un espacio físico con dicho terminal en el bolsillo, aunque también la misma persona podría utilizar otros terminales o dispositivos (ordenador de sobremesa, Tablet, o dispositivos similares).

Por otro lado, actualmente existen varios métodos (entre otros SMS, aplicaciones de mensajería instantánea como *whatsapp*, *email*, *push* de otras Apps, Push en la web, etc.) para enviar un mensaje que haga despertar (vibrar, sonido, activación de pantalla, etc.) al dispositivo móvil personal del usuario (ya la mayor parte de usuario lo tienen así configurado).

Sin embargo actualmente no podemos identificar que el mismo terminal de usuario o la misma persona que está visitando una página web o utiliza una APP, posteriormente visita un espacio físico determinado, y tampoco podemos hacer llegar un mensaje al terminal de usuario en el lugar adecuado con información relevante a su perfil (identificando su actividad online y offline simultáneamente: información de gustos, interés, hábitos, lugares que visita, frecuencia con los que lo visita, etc).

En el estado de la técnica actualmente conocido para la identificación online de los UTs y usuarios además del uso de cookies y seguimiento de direcciones IP que presentan muchos inconvenientes, existen publicados algunos métodos de "Browser fingerprinting" y de "App fingerprinting" que permitirían disponer de la identificación única del navegador o de la APP en terminal del usuario, pero son pocos los intentos en conseguir identificar por proximidad en el espacio físico esos mismos usuarios que pueden ser identificados de forma online, especialmente sin que el usuario tenga que sacar el terminal de usuario del bolsillo, sin que tenga que realizar acciones para ello cada vez que acude a un espacio físico, por ejemplo que abra una aplicación o el navegador buscando una promoción en tienda, o escuchan.

Actualmente los únicos sistemas que combinan la detección e identificación online y offline de UTs se centran en el uso de hardware muy simple tipo *beacon* bluetooth con protocolos (*Ibeacon* o *Eddystone*) que se basan en incluir cierta funcionalidad dentro de una APP (mediante un SDK *IBeacon*) o bien en el Navegador (si el usuario lo configura con los permisos adecuados) para escuchar los identificadores únicos que emiten los dispositivos tipo baliza denominados comúnmente en el arte por su nombre en inglés *beacon* y así identificar diferentes zonas del espacio físico desde la APP.

El problema es que además de la aplicación móvil o activar una configuración especial del navegador, se requiere la activación del bluetooth, y que el teléfono tenga un modelo de bluetooth compatible, estas condiciones deben cumplirse simultáneamente lo hace que el alcance de público de estos sistemas hoy en día sea muy restringido. En la práctica menos del 1% de los visitantes a una tienda reciben información por proximidad gracias a este sistema. Por otro lado, aunque la información que reciben los usuarios puede ligarse al comportamiento online en la APP en el caso del iBeacon, no hay ninguna relación con el comportamiento de ese mismo usuario cuando visita o interactúa en el Navegador con determinados sitios web como podría ser la tienda online o las redes sociales o la publicidad digital.

Por otro lado las técnicas de WiFi tracking basadas en la detección de la dirección MAC del dispositivo están diseñadas exclusivamente para identificar los UTs en una zona física, es decir para el análisis de comportamiento offline, sin posibilidad de ligarlo al comportamiento online del usuario, por ejemplo no están pensadas ni permiten en todos los OS la comunicación de mensajes *push* a la aplicación móvil cuando un terminal de usuario es detectado en un espacio físico.

Pero hoy en día, obtener un identificador único a partir de la dirección de la capa de acceso al medio de un terminal móvil en la actualidad es un problema que no está resuelto, por el hecho de que los sistemas operativos de los UTs actuales aleatorizan estas direcciones y un mismo dispositivos puede generar decenas de direcciones en un corto período de tiempo. La mayor parte de implementaciones del mercado no superan esta barrera de modo que al tratar de obtener información de dispositivos no asociados a un punto de acceso WiFi, la calidad de visitantes únicos reales que pueden identificarse es muy baja, ya que solo cuando el dispositivo esté conectado-asociado a los puntos de acceso WiFi disponen de un identificador único y offline del dispositivo.

Otra problemática conocida es el disponer de un identificador único del terminal de usuario y del usuario en offline y online, tanto para Android como para iOS, es otro problema que no está resuelto, ya que desde algunos sistemas operativos de amplio uso (como iOS 10) no está permitido a través de una APP (incluido el Navegador) el acceso vía software a los identificadores permanentes del dispositivo como podría ser la dirección MAC o dirección de acceso al medio de su tarjetas de red. En el caso de la aplicación se podría acceder al identificador de publicidad online del terminal de usuario, pero el navegador a su vez web no disponen de acceso al identificador de publicidad, y por otro lado cuando no está en uso el teléfono (es decir está offline) el identificador de publicidad online no se emite ni se comparte de ninguna forma, por lo que no puede utilizarse para cruzarlo con la detección offline de la dirección de acceso al medio de su tarjetas de red.

Se hace necesario por tanto disponer de la posibilidad de llevar a cabo una identificación única de dispositivos electrónicos, como los terminal de usuario, en *offline* y *online* que además permita distinguir y analizar el comportamiento offline no solo personas, sino también de cosas, por ejemplo una localización de sus activos.

DESCRIPCIÓN DE LA INVENCIÓN

A lo largo de este documento se hace uso de la siguiente terminologia:

- El término sistema para describir cualquier número de componentes, elementos, subsistemas, dispositivos, elementos de conmutación de paquetes, conmutadores de paquetes, conmutadores de acceso, enrutadores, redes, dispositivos o mecanismos de ordenador y / o de comunicación, o combinaciones de componentes de los mismos. El término ordenador incluye un procesador, memoria y buses capaces de ejecutar una instrucción en la que el ordenador se refiere a uno o a un grupo de ordenadores, ordenadores personales, estaciones de trabajo, mainframes o combinaciones de ordenadores de los mismos como puede ser una red. Asimismo se tiene que como usuarios se entiende cualquier persona física o cualquier sistema/maquina/robot software o físico que acceden o están presentes en cualquier tipo de:

a) espacio *online,* es decir que mantiene una conexión/sesión activa/abierta a través de cualquier tipo de red de comunicaciones o utilizan/interactúan con servicios online de información como páginas web o aplicaciones (en este documento referidas igualmente como APPs) o redes sociales).

b) espacio *offline* (tiendas, ciudades, centros comerciales, aeropuertos, etc..), es decir cuando estando presentes en el espacio físico no mantienen una comunicación/sesión activa ni están interactuando online, aunque disponen de algún un terminal (equipo con capacidades de procesamiento de información y comunicaciones) que les permite pasar a estar online en cualquier momento.

- NS Nodo Servidor: Hace de dispositivo de una arquitectura centralizada para almacenamiento, procesado y distribución de información al resto de elementos.

- Firmas: Sucesión o *array* de *fingerprints hasheados* (codificados) que genera un mismo dispositivo, según los IEs y/o las SSIDs recibidas, según las capacidades vía APP o vía WEB. Cada firma tiene una posición, las primeras se denominas firmas base, y se obtienen cuando se ha homologado y validado su deviceID, el resto son firmas adicionales que irán aportando valor en su conjunto.

- Visitante: persona o cualquier tipo de dispositivo (cosas máquinas, robots
- físicos o software) diferente de otros que accede a un espacio (físico o digital) generando eventos en dicho espacio.

- Visita, presencia corta continua considerada por la agrupación continua de eventos de un visitante en un espacio físico o digital.

- Visita, presencia agrupada de un visitante en un período considerando la agrupación discontinua de eventos de un visitante en un espacio físico o digital.

- Usuarios cualquier tipo de persona física o de cosa/maquina/robot software o físico que acceden o están presentes en cualquier tipo de espacio online (*website*, aplicaciones, redes sociales) o físico (entre otros viajeros de sistemas de transporte, ciudadanos, turistas, compradores en tiendas, y cualquier tipo de personas en cualquier tipo de espacio físico) que disponen de algún un dispositivo con capacidades de procesamiento de información y comunicaciones.

- Terminal de usuario , cualquier dispositivo llevado por una persona, activo/robot habitualmente en su vida offline, y que también es usado como dispositivo de acceso personal a medios online (*website*, aplicaciones, servicios, consultas, internet, etc... comunicaciones telefónicas, voz, audio, datos, etc..) que soporta múltiples aplicaciones, sensores, actuadores, pantallas, botones, teclados, etc. Los terminales de usuario pueden ser personales o no personales. Los dispositivos personales pueden hacer todas sus funciones aunque no esté bajo el control del humano, es decir cuando están en reposo o no activos. Pero se considera personal puesto que su entidad física sigue ligado a la persona física.

El objeto de la invención se basa en la utilización cotidiana de los dispositivos móviles que usamos en nuestro día a día (teléfono inteligente, tabletas, relojes inteligentes) que actúan de interfaz entre nuestra vida offline (cuando nos movemos a diversos sitios físicos con el teléfono en el bolsillo) y nuestra vida online (cuando nos movemos visitando páginas web o utilizamos aplicaciones móviles).

A diferencia del uso de sencillos sistemas emisores tipo *beacon* instalados en el espacio físico, junto con aplicaciones móviles que instalan y autorizan los usuarios para la escucha de los *beacon*, la invención descrita a continuación se basa en un dispositivo inteligente, que denominaremos Nodo, principalmente dedicado a la escucha y procesado de información radio de los UTs.

Ubicando este dispositivo inteligente Nodo en un espacio físico, éste recibirá y demodulará y decodificará las señales en las bandas públicas de frecuencia de (2,4 -3,6 y 5Ghz) en las que otros terminales móviles emiten frecuentemente debido a los protocolos de comunicaciones radio que utilizan (WiFi, Bluetooth, otros), por lo que analizando las tramas recibidas podrá detectar la presencia de dispositivos móviles en una zona física (espacio offline) de forma anónima sin intervención alguna.

En colaboración con la funcionalidad diseñada en forma de kit para los desarrolladores (Software Development Kit, SDK) utilizable en los sistemas operativos actuales (iOS, Android, ...) y también para HTML5, se construye un sistema identificador que denominaremos en adelante *Offline Online Identifier* (OOID) para lograr una alta probabilidad de identificar de forma única al terminal de usuario de forma única tanto en offline como online, lo que en adelante denominaremos OOID Homologado.

Se diseñan diversos mecanismos que propagan y actualizan la información contenida en el OOID a los diferentes elementos de la arquitectura (Nodo, APPs, Navegadores, Servidor). Dicho OOID persistirá y se irá actualizando en el terminal de usuario también cuando el terminal de usuario esté encendido pero no se use (offline).

Se aplica dicho dispositivo y métodos no solo a la generación de identificadores OOID, también para lograr el envío de mensajes por proximidad multicanal, consiguiendo alcanzar un alto número de UTs que sin ningún gesto del usuario reciben información asociada a su perfil en su terminal de usuario por proximidad en zonas físicas (siempre que lo hayan autorizado legalmente).

Se diseñan mecanismos que permiten que, de forma transparente al usuario, cuando el terminal de usuario pase cerca del Nodo se establezca una comunicación entre ambos, para el intercambio de información del OOID o el envío directo de mensajes al terminal de usuario.

Se diseñan métodos para maximizar el número de UTs que son identificados de forma única en el espacio físico por proximidad aunque utilicen direcciones MAC aleatorias o direcciones MAC Virtuales, con el objetivo de poder mejorar la identificación única de su comportamiento online y offline cruzado y poder enviar mensaje por proximidad a más UTs. En la práctica respecto al 1% del total de visitantes de dispositivos detectados de forma única que se

consiguen usando beacons bluetooth o el 20% que puede ser detectado utilizando balizas WiFi, se alcanzan con implementaciones del objeto de la invención y sistemas asociados tasas de entre el 60 al 80%.

El dispositivo y componentes asociados, permiten disponer de un sistema con muchas aplicaciones industriales hoy en día, gracias a disponer de un OOID, especialmente útil en el ámbito del marketing. En general en estas aplicaciones de marketing no es necesario garantizar al 100% la identificación única de los UTs en todo momento y en todos los entornos, sino solo tratar de comunicarse con el máximo número de personas posible en el lugar y momento apropiado, con mensajes relevantes para el consumidor que pueden elegirse analizando su comportamiento en varios medios o canales.

La información de comportamiento anónima online y offline que recoge el sistema es anónima, siendo imposible la identificación de la persona que lo porta, pero con el consentimiento del usuario puede ligarse a información personal o de otro tipo facilitada del usuario o disponible en otros sistemas (CRM, POS, etc..), el diseño incluye un novedoso sistema de creación de perfiles que permite incorporar información personal y no personal (psico-socio-demográfica, y transaccional) mezclada información de comportamiento anónima online y offline (en App, Web y localización física) obtenida 24/7 y en tiempo real.

DESCRIPCIÓN DE LOS DIBUJOS

Para complementar la descripción que se está realizando y con objeto de ayudar a una mejor comprensión de las características de la invención, de acuerdo con un ejemplo preferente de realización práctica de la misma, se acompaña como parte integrante de dicha descripción, un juego de dibujos en donde con carácter ilustrativo y no limitativo, se ha representado lo siguiente:

Figura 1.- Muestra un diagrama donde se aprecia un posible modelo de arquitectura y la enumeración de los elementos considerados y la dirección de los principales flujos de mensajes o información entre ellos.

Figura 2.- Muestra un ejemplo de tabla de eventos donde se detalla la estructura y funciones necesarias en los eventos entre elementos del sistema

Figura 3.- Muestra una implementación del OOID en la cual se detalla la estructura y funciones necesarias en el identificador *online* y *offline*.

Figura 4.- Muestra un diagrama donde se aprecian los procesos para el para verificar el OOID.

Figura 5.- Muestra un diagrama donde se aprecian los procesos dedicados a la propagación del OOID.

Figura 6.1.- Muestra una tabla descriptiva del Dispositivo Nodo.

Figura 6.2.- Muestra un esquema del sistema de Notificaciones directas de nodo

Figura 7.- Muestra una tabla en la cual se aprecian las principales funciones del SDK APP.

Figura 8.- Muestra una tabla en la cual se aprecian las principales funciones del SDK WEB.

Figura 9.1.- Muestra una tabla en la cual se aprecian las principales funciones Nodo Servidor o Servidor.

Figura 9.2.- Muestra un diagrama donde se aprecia el funcionamiento del análisis del comportamiento, detallando el módulo de gestión del comportamiento *online* y *offline*.

Figura 9.3.- Muestra un esquema donde se aprecian los parámetros del cómputo de visitantes, visitas y visita es describiendo las técnicas de análisis del comportamiento online y offline.

Figura 9.4.- Muestra un diagrama donde se aprecia la interacción entre sistemas, detallando las operaciones entre módulos y subsistemas.

REALIZACIÓN PREFERENTE DE LA INVENCIÓN

En una posible realización preferente del objeto de la invención se requiere disponer de identificadores únicos que permitan cruzar la información de comportamiento offline y online, se hace uso los siguientes elementos que se describen en la arquitectura de la figura 1.

En diferentes realizaciones alternativas podrían variarse configuraciones de la arquitectura que se describe a continuación, y que ha sido considerada como mejor alternativa en el estado actual de la tecnología, pensando en utilizar un elemento central o servidor y un amplio número de dispositivos de bajo coste (nodos) formando una extensa red, aunque también se podría utilizar un modelo donde las funciones del servidor están distribuidas en los nodos teniendo un escenario de red distribuida.

En esta realización preferente los elementos que se combinan en la arquitectura son:

- Un dispositivo hardware inteligente o Nodo (101), que comprende entre otros con un módulo escucha de radiofrecuencia.
- Una serie de dispositivos electrónicos (102) tales como equipos terminales de usuario (102), referidos en partes de este documento como UTs (102), capaces de ejecutar aplicaciones móviles donde se integra un SDK APP (104) para aplicaciones móviles o APPs, y donde corre un navegador web que a su vez tienen integrado un SDK web (105) preferiblemente bajo HTML5 para páginas web.
- NS o Nodo Servidor (103) con subsistemas o módulos– preferiblemente implementado en hardware software- de análisis y propagación de identificadores *online* y *offline* (106) , módulo gestión del comportamiento (107) offline y online y un módulo gestor de campañas que incluye un módulo de análisis de perfiles y resultados (108) , todo ello con diferentes conexiones entre ellos. El Nodo Servidor (103) actúa a como centralizador.

- NS o Nodo Servidor (103) preferiblemente implementado en hardware, que incluye entre otros subsistemas o módulos–:

  a) software- de análisis y propagación de identificadores *online* y *offline* (106)

  b) módulo gestión del comportamiento (107) offline y online

  c) un módulo gestor de campañas que incluye un módulo de análisis de perfiles y resultados (108) , todo ello con diferentes conexiones entre ellos.

- El Nodo Servidor (103) actúa a como centralizador.

A través del módulo de análisis del comportamiento (107) se puede obtener información de las detecciones *offline* (eventos de nodo), que son analizadas para calcular visitantes, visitas y visita es, y actualiza datos de un Identificador Online Offline, denominado OOID por sus siglas en inglés, al resto de elementos.

A través del módulo de análisis de comportamiento (107) se obtiene información (eventos de SDK) del uso online de aplicaciones móviles del SDK APP (104) y navegadores visitando páginas web mediante el SDK WEB (105) para poder hacer un análisis del comportamiento online cruzado en el módulo de análisis de perfiles y resultados (108).

A través del módulo de análisis de perfiles y resultados (108) se generan perfiles de comportamiento cruzados *offline* y *online*, y analiza los resultados de campañas de comunicación, actualizando perfiles y los identificadores OOID de los UTs (102).

A través del módulo de campañas se intercambia información entre los diversos elementos de la arquitectura para hacer llegar un mensaje tipo *push* a través de los diversos canales posibles habilitados hacia el terminal de usuario (102) del usuario.

Por otro lado se diseña un subsistema de comunicación basado en eventos (tramas o piezas discretas de información) desde los UTs (102)y Nodo/s (101) hacia un servidor (SETS, *Send Events to Server*) y un protocolo de comunicación desde el servidor al Nodo y UTs (REFS: *Receive Elements from Server*).

En varias implementaciones para cualquier comunicación entre los elementos de la arquitectura se podría utilizar eventos de Nodo (aquellos que viajan entre el Nodo (101) y el nodo servidor (103) o eventos de SDK (aquellos que viajan entre los SDK y el nodo servidor (103), con las siguientes elementos entre otros:

- *Timestamp*, marca de tiempo universal del momento del evento APP_ID, identificación de una aplicación digital o espacio físico.
- *Node_ID*, identificación un nodo o una zona física.
- *Geoloc*, coordenadas y precisión de geolocalización.
- *Type* evento o información que se envía y sus parámetros específicos.
- *MoreInfo*, puede utilizare para enviar más información de mantenimiento, envio de listas, objetos, etc.

El identificador OOID puede comprender entre otros los elementos que se aprecian en la figura 3. De esta manera se tiene un Identificador de dispositivo (dID), que se puede generar el a partir de un Hash SHA1 en tiempo de ejecución de la dirección de la capa de acceso al medio (sea o no aleatoria) que puede obtenerse en las tramas escuchadas desde el Nodo (101), o vía software en ciertas versiones del OS de los UTs (102).

Aunque el resto de elementos presentes en la figura 3 y que se describen a continuación no se incluyeran, el dID estaría siempre presente en el OOID aunque sea con un valor aleatorio, no homologado, a su vez el dID podrá ser actualizado por el proceso de propagación de OOID por los SDKs (104, 105), el nodo servidor (103) o el Nodo (101) en el momento que se conozca el OOID homologado.

Tal y como se desprende de la figura 3 el OOID puede comprender un OUI, que se corresponde con los tres primeros bytes de la dirección de la capa de acceso al medio del terminal de usuario (102) y se utilizará en el proceso de homologación del OOID.

El objeto de la invención contempla la implantación de una sucesión de firmas que denominamos por su término en inglés *array* o en español sucesión de Firmas, que permiten ir añadiendo firmas al mismo, entendidas dichas firmas como características propias del terminal de usuario (102) o de su comportamiento, que pueden ser obtenidas y propagadas por Nodos (101), Nodo Servidor (103), SDK APP (104) y SDK WEB (105) de forma combinada. Es decir, se obtienen combinando fuentes de información *online* y *offline*.

El objeto de la invención hace uso de procesos de homologación, unificación, aceptación y propagación de OOIDs. En algunas implementaciones se podrían realizar los siguientes procesos en cualquiera de los componentes de la arquitectura que reciben eventos con OOIDs de UTs (102), especialmente en el Nodo Servidor (103); para ello en la homologación se puede marcar con el estatus de Homologado aquellos OOIDs que por cualquier método se ha comprobado que su dID proviene de una dirección real y no aleatoria de la capa de acceso al medio del terminal de usuario (102). Se utilizan para determinar el número de visitantes únicos y para tener identificación única en el sistema de envío de mensajes.

En algunas implementaciones los OOID Homologados pueden ser obtenidos de forma directa:

- Por detección offline del Nodo a partir de la dirección de la capa de acceso al medio real de las tramas escuchadas tanto cuando el dispositivo está descubriendo tramas como cuando está asociado a un punto de acceso, en adelante AP.
- Por detección del Nodo (101) cuando el navegador se conecta un UT (102) a un portal cautivo del Nodo (101).
- En el SDK APP (104) Android por software a partir de la dirección de la capa de acceso al medio.
- En el SDK APP (104) iOS hasta iOS9 por software a partir de la dirección de la capa de acceso al medio.
- En el SDK APP (104) iOS por software cuando el Smartphone tiene conectividad WiFi hasta 10.3 a partir de la dirección de la capa de acceso al medio.
- A partir del iOS 10.3 se combinan diferentes técnicas como la conexión directa al punto de acceso- AP oculto del Nodo por Proximidad o la propagación del OOID entre los diferentes elementos de la arquitectura.

En una posible realización alternativa podría homologarse el OOID al generar el dID en los SDK (104, 105) y Nodos, o posteriormente en el Nodo Servidor solo con el OUI sin necesidad de almacenar o enviar al nodo servidor (103) la dirección real completa (por privacidad), para ello se podría contrastar contra la lista de OUI de fabricantes oficial, contra un histórico de OUIs reales (usadas por fabricantes chinos que no utilizan la lista oficial) disponible antes de que existieran los procesos de aleatorización, y contra una lista de OUI virtuales o falsas, que han sido detectadas como tales, ya que aparecen con un comportamiento no natural, incluso en horario nocturno, y comprobados con los diferentes modelos de UTs (102) en diferentes ubicaciones.

En la unificación se puede marcar el estatus de Unificado de un OOID no homologado cuando haya un alto grado de coincidencia o pequeña distancia entre el conjunto de firmas con otro OOID Homologado (que se ha recibido a través el proceso de propagación. De esta manera se puede crear un registro de OOIDs unificados entre ellos que se utilizaría entre otras tareas para determinar el número de visitantes únicos y visitas en el proceso de análisis de comportamiento *offline*. Cuando un dispositivo se unifica, entre otros procesos, podrían agregarse las firmas coincidentes entre el OOID homologado y al OOID no homologado, y propagarse así un OOID unificado en firmas. En este proceso de unificación podrían eliminarse firmas repetidas o contenidas en otras para optimizar el proceso. Y podrían ordenarse de forma que la medición de coincidencia pudiera implementarse

ágilmente en el caso de muchas firmas, asignando un peso mayor a las primeras firmas que permiten identificar mejor de forma única a un terminal de usuario (102).

Se podría medir la distancia entre dos firmas de dos OOID (A y B), y la coincidencia entre firmas de los OOIDs según las siguientes fórmulas:

$$Distancia\ D = (firmaA - firmaB \times peso\ firma)$$

donde el *peso firma* es un valor configurable que puede modificarse para dar un ajuste fino al sistema. Y la coincidencia entre el conjunto el firmas de cada OOID, como la suma de las distancias de las firmas comparadas una a una para aquellas existentes entre ambos OOIDs,

$$Coincidencia\ C = Sum\ i = 1; i = n\{ D(i) \}$$

Donde *n* es el número de firmas coincidentes disponible entre ambos OOIDs.

Para la aceptación se puede marcar el estatus de Aceptado de un OOID no homologado y no unificado cuando hay un grado de coincidencia medio entre el conjunto de firmas con otro OOID Homologado. Asimismo podría crearse un registro de OOIDs unificados entre ellos que podría utilizar en el sistema de comportamiento para determinar el tiempo de presencia el número de visitas. En este caso no tendría que producirse la concatenación de firmas ni la propagación. También podría crearse un registro de OOID unificados entre ellos.

Para la validación se puede marcar el bit de Validado (y por tanto podrían ser utilizados para el análisis y otros procesos) de aquellos OOIDs que no están excluidos del por diferentes motivos, entre otros:

- Porque se encuentran en listas negras (Robinson).
- Porque han sido excluidos por otros métodos de verificación de datos válidos, como:
    - Aparecen simultáneamente en dos ubicaciones muy lejanas.
    - No tienen suficientes detecciones con variaciones de potencia que implican un cierto movimiento dentro un espacio con varios nodos (101), es decir su comportamiento es similar a dispositivos fijos.

Una parte del método objeto de la invención se basa en la identificación online y offline a partir de la propagación de los OOID obtenidos para los UTs (102), por varias vías (como

eventos o notificaciones *push* silenciosas) y se aplica, no solo al nodo servidor (103) sino a todos los elementos de la arquitectura, implementando al menos las funciones que se aprecian en la figura 5.

De esta manera se tiene que Al recibir un OOID, éste se compara con los OOID disponibles, siendo actualizado si procede según los métodos de homologación, unificación, aceptación y validación. Normalmente los SDK (104,105) que corren los UTs (102), tienen información solo de su OOID, pero el nodo servidor (103) recibirá en algún momento todos los OOID generados en el sistema, los nodos (101) podrían recibir información de todos los OOID o solo de parte de ellos (los que están en la misma zona NodeID o aplicación APPID).

El nodo servidor (103) puede difundir los OOID homologados a otras appplicaciones y Navegadores, mini-navegador (que se lanza automáticamente en algunos sistemas operativos de UTs (102) cuando se accede a un portal cautivo) y navegadores embebidos (*webviews*) mediante diversas técnicas entre otras:

- Mediante *push* silenciosas (de forma activa).
- Mediante funciones del SDK cada vez que se arranca o corre un
- proceso de la APP.
- Mediante funciones del SDK cada vez que se conecta a una página con SDK HTML5

De esta manera, el proceso de propagación garantiza la propagación de OOID Homologados:

- Recibidos de un *WebViewBrowser* que a su vez lo ha obtenido:
    - Al ser abierto por una APP
    - De una *cookie* compartida.
- Recibidos desde el Navegador que a su vez lo ha obtenido:
    - Actualizado por el nodo servidor (103).
    - Actualizado por el Portal Cautivo del Nodo (101).
- Recibido del servidor, que a su vez lo ha obtenido de:
    - Otra App que ya lo ha obtenido previamente desde:
        - El MiniNavegador, Navegador con Conexión a un Portal Cautivo del Nodo (101)
        - El *WebViewBrowser.*

Al propagar un OOID se puede codificar el número del status de propagación, para conocer si ha sido difundido varias veces a todos los elementos de la arquitectura. Solo los OOID homologados/unificados y validados se propagan, actualizándose en tiempo real. El nodo servidor (103) puede actualizar es estado de los OOID que no están Homologados a Homologados y también actualizar su valor en los históricos de datos asociados a UTs (102), o solo almacenar la relación histórica entre OOIDs a partir de la fecha de homologación sin alterar los OOID de datos pasados.

De esta manera se tiene un sistema cuya arquitectura y configuración permite llevar a cabo las acciones anteriores para poder implementar, en una posible realización del mismo, el método objeto de la invención. Dicha arquitectura correspondiente a una realización preferente del objeto de la invención comprende el dispositivo Nodo (101), que responde a una estructura como la mostrada en la figura 6.1 y que según implementaciones puede incluir entre otras las siguientes funcionalidades etiquetadas en dicha figura 6.1, teniendo de esta forma las siguientes funcionalidades:

- SETS: Escuchar y procesar tramas que emiten los UTs (102) tanto cuando están como cuando no están asociados a redes inalámbricas que hacen uso de WiFi, es decir tipo WLAN. A partir de la información de dichas tramas podrían generar en tiempo de ejecución un OOID, procesarlo entre otros con los sistemas de homologación, unificación, aceptación y validación descritos anteriormente, y generar uno o varios eventos de Nodo (101). Podrían escuchar y procesar tramas que emiten los UTs (102) cuando disponen de comunicaciones *bluetooth* (BT). A partir de la información de dichas tramas pueden generar en tiempo de ejecución un OOID, procesarlo entre otros con los sistemas de homologación, unificación, aceptación y validación descritos anteriormente, y generar uno o varios eventos de Nodo (101). Entre otros mecanismos se considera establecer una red privada virtual (referida en este documento como VPN por sus siglas en inglés) con el servidor que garantice la privacidad de la información entre ambos. Los eventos de Nodo generados pueden ser almacenados hasta que la conexión con el servidor esté disponible.] Para ahorrar cómputo en los sistemas de comportamiento, se prevé poder agrupar durante una ventana de tiempo pequeña los datos de las tramas detectadas o detecciones realizadas y enviarlas en un único evento de Nodo, que continene además del OOID solo la información del número de detecciones y la potencia media de detección. Por tanto los eventos podrían contener información de

una sola detección o varias en un corto espacio de tiempo. Adicionalmente en cada evento de nodo podría incluir entre otros datos más información útil para el mantenimiento remoto (como su dirección IP, coordenadas GPS, lista de SSiD cercanas).

- REFS Nodo: Entre otras informaciones podría recibir en tiempo real del servidor, información de OOIDs homologados, para la ejecución del proceso de unificación. Entre otras informaciones para dar soporte a toda la funcionalidad del sistema podría recibir en tiempo real del servidor, información de eventos de otros Nodos (101), OOIDs de UTs (102) que entran en Nodo Virtual *geofence* con sus SSIDs, campañas, perfiles, *tokens* de comunicación con los terminal de usuario (102), la lista de OUI de fabricantes, un histórico de OUIs disponible antes de que existieran los procesos de aleatorización, una lista de OUI virtuales o falsas y la lista de OUI validados.
- Punto de acceso Oculto NODO: Se puede generar un punto de acceso (AP) oculto solo conocido por los UTs (102) donde se haya integrado el SDK APP (104), permitiendo conexiones WLAN con dichos equipos UTs (102). Asimismo puede generar un punto BT modo visible para parear con otros terminal de usuario (102) que conozcan un número de enlace (PIN) solo disponible en los UTs (102) donde se haya integrado el SDK APP (104), permitiendo conexiones BT con dichos UTs (102). En algunas implementaciones podría ser mediante socket y que podría entre otras informaciones enviar su OOID Homologado a los UTs (102) conectados, ya que dispondría en la conexión de nivel 2 de la información suficiente para Homologar el OOID. Pudiendo desconectarles una vez se reciba confirmación de que se ha recibido correctamente toda la información en el UT(102) para ser eficaz en el número de terminal de usuario (102) conectados simultáneamente.

Un Portal Cautivo Nodo es el encargado de poder generar un punto de acceso, un AP, visible en modo abierto con un portal cautivo, para permitir conexiones con cualquier usuario que lo desee (por ejemplo porque está intentando registrase en un programa de fidelización, validar un cupón recibido por email, o porque está buscando acceso gratuito a internet). Mediante técnicas de modificación dinámica del DNS del nodo (101) u otras del portal cautivo podría redireccionarlo siempre al navegador del terminal de usuario UT (102) a un mismo dominio (el asociado al servidor web del NS por ejemplo) cargando así la página web principal del portal cautivo del Nodo que incluiría el SDK WEB (105). De esta manera Nodo (101) identificaría y accedería a las tramas enviadas para la conexión al portal cautivo,

podría entre otros métodos homologar el OOID y mediante el SDK WEB (105) almacenarlo en un dispositivo almacenamiento local que denominamos por su nombre inglés *localstorage* o en forma de cookies asociadas a dicho dominio predeterminado..

Un nodo no autorizado tipo *Rogue* que deminamos Rogue Nodo puede ser el encargado de generar tramas *beacon* WLAN para anunciar un SSID disponible y obligar a los terminal de usuario (102) cercanos a enviar tramas de asociación si tienen dicho SSID en su lista de redes conectadas. Entre otras fuentes la lista de SSID a anunciar podrían provenir de la recibida del servidor y obtenida previamente por los SDK instalados en las APPs de los UTs, de los SSID más populares pero no detectados en la cercanía por el Rogue Nodo (puesto que estos ya se anuncian) o del histórico de SSID anunciados en las tramas de descubrimiento WLAN de los UTs que son incluidos en las firmas de los OOID intercambiados con el Nodo Servidor (103) (provenientes de otros nodos). El mecanismo para priorizar, optimizar y disparar el anuncio de SSIDs podría ser entre otros anunciar primero y periódicamente, hasta recibir la respuesta del terminal de usuario (102) en un plazo de tiempo configurable, aquellos SSID provenientes del SDK APP (104) de UTs (102) que conocido por geolocalización (que podemos llamar Nodo Virtual) han entrado en unas coordenadas dentro de radio configurable alrededor del Nodo (por ejemplo 0,5km).

Un Nodo Beacon puede ser el encargado de emitir tramas por bluetooth según los protocolos iBeacon ™ y EddyStone ™ que a través de una APP o el Navegador podría ser detectado y a través del SDK APP (104) y podría generar un evento de Nodo (aunque no haya sido generado por el nodo sino por el UT (102) con la información del OOID Homologado si fuera conocido por el UT (102).

De manera opcional se puede hacer uso de un nodo de Push directo que permite incluir un gestor de campañas dentro del propio nodo (101), evitando los retrasos en el envío de mensajes desde servidor, y haciendo que sea el propio nodo (101) el que gestione el envío de los mensajes hacia el terminal de usuario (102), o incluso cuando dispone de comunicación directa con el terminal de usuario (102) por proximidad a través del sistema diseñado para tal fin, puede llamar a funciones directas de los SDK (104, 105) para mostrar al usuario el mensaje directamente (mediante entre otras técnicas como notificaciones *push* locales, PopUps, etc..) sin necesidad de utilizar los canales disponibles del terminal de usuario (SMS, Push, Email, ...). El Nodo recibiría al Nodo Servidor (103) los tokens de los OOIDs homologados así como cualquier otra información relativa a las campañas.

Podría utilizar entre otras técnicas el ofrecer mediante el protocolo *HotSpot2.0* para proporcionar un canal de comunicación y que los UTs (102) utilicen este canal para preguntar a un AP por la información de acceso a la red usando el ANQP (*Access Network Query Protocol*) con su dirección de la capa de acceso al medio real, aumentando así la probabilidad de obtener el OOID homologado en menos tiempo para aquellos dispositivos que la aleatorizan. Podría utilizar entre otras técnicas para aumentar la probabilidad de obtener el OOID homologado en menos tiempo para aquellos dispositivos que aleatorizan la dirección de la capa de acceso al medio, *HotSpot* 2.0 HS2.0 para proporcionar un canal de comunicación y que los UTs (102) utilicen este canal para preguntar a un AP por la información de acceso a la red usando el ANQP (*Access Network Query Protocol*) ofreciendo su dirección de la capa de acceso al medio real. De esta manera se podría utilizar, entre otras técnicas, el anunciar tramas del protocolo utilizado para facilitar el emparejamiento entre dispositivos llamado Wi-Fi Protected Setup (WPS), para forzar a los UTs (102) a generar un identificador UUID, que será recogido como una firma más que ayudará a unificar los UTs (102) que aleatorizan la dirección de la capa de acceso al medio real.

Para aumentar la privacidad en la localización e identificación frente a terceros de los UTs (102) que están en el espacio físico de alcance del Nodo (101) entre otros mecanismos, el nodo (101) puede generar en tiempo real o de ejecución tramas de descubrimiento de las redes WLAN o BT, que normalmente son emitidas por los UTs (102), usando para ello las mismas direcciones de la capa de acceso al medio de los UTs (102) escuchados recientemente, o de los escuchados por otros nodos y que han sido recibidas por el proceso de propagación, y también generarlas aleatoriamente con diferentes patrones comunes similares a los recibidos habitualmente.

Asimismo se contempla el poder realizar el análisis de SSIDs recibidos en las tramas de descubrimiento entre otras, generando una firma por cada SSID-S detectado, que pueden concatenarse para obtener mayor eficiencia en el proceso de unificación de OOIDs. Pudiento realizar el análisis de los Elementos de Información (IEs) contenidos en las tramas WLAN como son entre otros Supported Rates, High Throughput capabilities , Interworking Capabilities, y WPS o Wi-Fi Protected Setup , podemos disponer de un alto número de dispositivos únificados para ciertos modelos, y aportan la base de firma para el resto de modelos. En concreto a través de la firma IEs podemos distinguir si es un dispositivo iOS8 o superior, lo que ayuda a optimizar procesos de Unificación, Aceptación y Validación.

También se contempla la posibilidad de realizar el análisis de las semillas barajadas y predictivas de las propias radios utilizadas por los UTs (102) para detectar coincidencias.

En aquellas realizaciones donde se hace uso del SDK APP (104) se hace uso de las funcionalidades del SDK APP (104) que se muestran en la figura 7 y que responden a:

SETS:

Podrían generar un OOID a partir de la información disponible vía software, y entre otros momentos cada vez que arranca o antes de cerrar la aplicación o APP o en el momento que hay cualquier interacción del usuario, se podrían generar eventos de SDK APP (104) hacia el Nodo Servidor (103). Para obtenerse el OOID homologado el SDK APP (104) podría acceder a la dirección de la capa de acceso al medio del terminal de usuario (102) y en caso de que no sea accesible se generaría un dID aleatorio y un OOIDs no homologado. Al recibir información desde el Nodo Servidor (103) se actualiza el OOID con el OOID homologado en *localstorage*, cookies u otros medios para asegurar la persistencia de la información accesible al SDK APP (104).

Seguimiento de objetos y uso:

Para conocer más sobre los gustos, intereses, etc. se incorporan funciones que el programador podría utilizar en su APP para el seguimiento del comportamiento de objetos y del propio uso de la misma APP por parte del usuario a través de firmas obtenidas de:

- Eventos de Sesión SESION-S, cada vez que el usuario abre la APP podria incrementarse en uno el número de sesiones realizadas por la APP en ese terminal de usuario (102).
- Eventos de clicks CLICK-S, cada vez que el usuario hace click, podría acumular el número de click sobre cada objeto a analizar en un periodo
- Eventos de texto TEXT-S, cada vez que el usuario escribe en un campo o el programador pasa un parámetro de caracteres alfanumérico (por ejemplo los puntos acumulados en el juego o programa de fidelización o la apertura o cierre de la aplicación).
- Eventos de transacciones electrónicas ECOMM-S, similares a eventos de texto pero con parámetros específicos que identifican las ordenes de pedido en tiendas electrónicas.

- Eventos de impresión visual de objetos IMPR-S, donde se recoge el tiempo total y número de veces que un objeto ha estado visible en pantalla del terminal de usuario (102), teniendo en cuenta el espacio en pantalla.
- Eventos de Navegación NAVI-S, que recogen la profundidad de navegación indicando si ha pasado por varias secciones de la APP antes.
- Eventos multimedia MDIA-S, que recogen la configuración actividad con reproductores, por ejemplo si ha terminado de ver el video.

Tokens y datos personales:

Entre otros datos, se pueden enviar al Nodo Servidor aquellos datos - personales o no personales- que hayan sido suministrados por el usuario y recogidos a través de las funcionalidad de seguimiento del comportamiento y la interacción que dispone, que pueden servir como *tokens* de comunicación (es decir identificadores en un sistema de comunicación) entre otros datos como número de teléfono o email, o *login* sociales que podrían servir de tokens para el envio de mensajes.

Rogue:

Entre otros datos podría enviar al Nodo Servidor (103) las SSID con las que se ha conectado el terminal de usuario (102) que llegarán al Nodo (101).

REFS:

Entre otros datos, se puede recibir en tiempo real del Nodo Servidor (103), información de definición de zonas geolimitadas o *geofence* por su nombre en inglés, información de campañas como sus elementos y objetos, la lista de OUI de fabricantes, un histórico de OUIs disponible antes de que existieran los procesos de aleatorización, y una lista de OUI virtuales o falsas, las lista de OUI validados. Al recibir notificaciones *push* externas podría gestionar la recogida de parámetros de la notificación, para que automáticamente según la programación de la campaña pudiera abrir navegadores embebidos en la APP o navegadores externos.

Geofence:

Se puede generar un evento de tipo Nodo (como si le hubiera detectado un nodo por proximidad), para disponer de un Nodo virtual en cualquier punto físico, y para la activación de las funcionalidad de rogue Nodo en el Nodo (101).

Interrogation:

Se puede conectar al punto de acceso AP oculto de los nodos o a la conexión BT ofrecida por estos, y a través de socket u otros procedimientos de comunicaciones intercambiar entre otros datos su OOID homologado. Será en el proceso de instalación de la APP que se pediría al usuario los permisos oportunos para la conexión directa al AP oculto de los nodos o incluso la instalación de un perfil para tal fin si fuera necesario en algunos modelos de sistemas operativos de ETs.

Propagación:

Entre otros momentos cada vez que conoce su OOID homologado se puede propagar al Nodo servidor (103) así como a los navegadores del UT (102), para ello podría utilizar entre otros métodos los siguientes:

- Abrir un navegador embebido en la APP (p.e.WebView) y pasarle como parámetros en la URL el OOID homologado.
- Abrir un navegador externo configurado por defecto en el terminal de usuario y pasarle como parámetros en la URL el OOID homologado.

PUSH Directo:

A través de un gestor de campañas interno como el descrito anteriormente se puede analizar frecuentemente la lista de campañas y objetos recibidos para lanzar una Notificacion Push Local, evitando la necesidad de que tenga cualquier conexión a internet para recibir mensajes push en el momento programado en la campaña.

INTERNAL_AP:

En aquellos UTs (102) que su OS / sistema operativo lo permitiera (con los permisos de usuario adecuados) podría genera un AP oculto, mientras que el UT (102) no utiliza la conexión WLAN, al que puede conectarse cualquier otro UT (102) -incluso el mismo- obteniendo el SUID Homologado del UT (102) de forma totalmente análoga a como se realiza en el nodo (101) por este medio.

En el proceso de instalación de la SDK APP (104) el usuario aceptará las condiciones de uso y dará todos los permisos oportunos o incluso instalará los perfiles de uso necesarios, para cada una de las funcionalidades que se han descrito anteriormente, asegurando el cumplimiento de la legislación en esta materia.

Privacidad y permisos:

En el proceso de instalación de la aplicación APP con el SDK APP (104) integrado es el usuario quien autoriza todos los permisos y acepta las condiciones de uso que se derivan de toda las funciones antes enunciadas

Firmas:

Se establece la posibilidad de utilizar capacidades u otros identificadores obtenidas desde la APP via APIs del OS, como son entre otras el IDFA/GAID, Carrier, Resolucion de pantalla y profundidad de color, Modelo, Memoria, Apps instaladas para generar otras firmas. En concreto podría generarse la firma IDFA/GAID, ya que este es un identificador único por dispositivo accesible solo vía software desde una aplicación móvil. Todas las aplicaciones móviles de un mismo dispositivo tienen el mismo GAID/IDFA proporcionado por el fabricante, para uso publicitario. Asimismo, podrían utilizarse entre otros Genero, Fecha de Nacimiento o Código Postal facilitados por el usuario directamente en la interacción con la APP, para generar una firma basada en datos personales PIIS (Firma de información Personal Identificable). Dicha información será suministrada y tratada por el sistema de acuerdo a la legislación vigente. Es decir, por ejemplo, no se generarán estas firmas sin el consentimiento informado y previo del usuario, y podrán eliminarse.

También se prevé utilizar otros datos no personales facilitados por el usuario directamente en la interacción con la APP, para generar una firma basada en datos no personales NPIS (Firma de información No Personal). Dicha información será suministrada y tratada por el sistema de acuerdo a la legislación vigente.

En aquellas realizaciones en las que se hace uso del SDK WEB (105) se tiene que éste influye las funcionalidades que se aprecian en la figura 8 y que corresponden a:

SETS:

Entre otros momentos cada vez que carga el sitio web con el SDK HTML5 o antes de cerrar dicha página podría generar eventos de SDK que serán enviados al Nodo a través de llamadas a funciones del NS (por ejemplo funciones de Php u otras plataformas de aplicaciones de Servidor). Al recibir información o al ser lanzado desde una APP actualizar su OOID, almacenado en *localstorage*, *cookies* u otros medios para asegurar la persistencia de la información accesible al SDK. Al conectarse al Portal cautivo del nodo (101) se

actualiza el OOID almacenado en *localstorage* o en *cookies* u otros medios para asegurar la persistencia de la información accesible al SDK WEB (105).

Funciones de seguimiento de objetos:

Para conocer más sobre los gustos, intereses, etc. se pueden incorporar funciones que el programador podría utilizar en su código HTML para el seguimiento del comportamiento de objetos y del propio uso del sitio o aplicación WEB por parte del usuario a través de firmas obtenidas de:

- Eventos de Sesión SESION-S, cada vez que el usuario abre la aplicación WEB podría incrementarse en uno el número de sesiones realizadas por la aplicación WEB en ese terminal de usuario (102).
- Eventos de clicks CLICK-S, cada vez que el usuario hace click, podría - acumular el número de click sobre cada objeto a analizar en un periodo.
- Eventos de texto TEXT-S, cada vez que el usuario escribe en un campo o el programador pasa un parámetro de caracteres alfanumérico (por ejemplo los puntos acumulados en el juego o programa de fidelización o la apertura o cierre de la aplicación).
- Eventos de transacciones electrónicas ECOMM-S, similares a eventos de texto pero con parámetros específicos que identifican las órdenes de pedido en tiendas electrónicas.
- Eventos de impresión visual de objetos IMPR-S, donde se recoge el tiempo total y número de veces que un objeto ha estado visible en pantalla del terminal de usuario (102), teniendo en cuenta el espacio en pantalla.
- Eventos de Navegación NAVI-S, que recogen la profundidad de navegación indicando si ha pasado por varias secciones del sitio o aplicación WEB antes.
- Eventos multimedia MDIA-S, que recogen la configuraciónactividad con reproductores, por ejemplo si ha terminado de ver un video, etc.

Tokens y datos personales:

Entre otros datos podría enviar al Nodo servidor (103) aquellos datos personales o no personales que hayan sido suministrados por el usuario y recogidos a través de las funcionalidad de seguimiento del comportamiento y la interacción que dispone, entre otros datos como número de teléfono o email, o datos de acceso a redes sociales que pueden servir de tokens para el envío de mensajes a los UTs (102).

PUSH Web:

Para implementar las denominadas notificaciones *push* web, al recibir el SDK web (105) un evento del NS en tiempo real via AJAX o cualquier otro protocolo web de envío de eventos asíncronos, podría abrir otras ventanas (pop-up) o cambiar dinámicamente el contenido de un banner en la página web donde se integra o cualquier otra acción que desee integrar el programador en la página web.

Notificaciones de navegador:

Se prevé utilizar el sistema de notificaciones propio del navegador (según el tipo de navegador abierto) si cuenta con los permisos apropiados.

Firmas:

Puede obtenerse una firma WEB-S en forma de *array* de firmas a su vez a partir de recopilar información del propio terminal de usuario (102) según las capacidades obtenidas desde WEB, como son entre otras *UserAgent*, *http accept header*, o la resolución de pantalla y profundidad de color entre otras. Podrían utilizarse entre otros Genero, Fecha de Nacimiento o Código Postal facilitados por el usuario directamente en la interacción con el sitio WEB que son recopilados a través de las funciones de seguimiento de objetos del SDK web (105), para generar una firma basada en datos personales PII-S (Firma de información Personal Identificable). Dicha información será suministrada y tratada por el sistema de acuerdo a la legislación vigente. Es decir, por ejemplo no se generan estas firmas sin el consentimiento informado y previo del usuario. Se pueden utilizarse otros datos no personales facilitados por el usuario directamente en la interacción con el sitio WEB, para generar una firma basada en datos no personales NPI-S (Firma de información No Personal). Dicha información será suministrada y tratada por el sistema de acuerdo a la legislación vigente.

Propagación:

Entre otros momentos cuando el usuario lanza el navegador del terminal de usuario (102), cada vez que accede a un sitio o aplicación WEB, se podría cargar el SDK WEB (105) y mediante técnicas de paso de parámetros en le URL u otras técnicas, se intercambiaría el OOID Homologado con el NS. El OOI se guarda en el *localstorage* del Navegador o en *cookies* si es posible, entre otros métodos. También al lanzarse el navegador embebido (tipo WebView) desde una APP con el SDK APP (104) se intercambia el OOID homologado con la APP. El OOI se guarda en el *localstorage* del Navegador o en *cookies* si es posible, entre

otros métodos. Análogamente, al lanzarse el navegador externo configurado por defecto en el terminal de usuario (102) y pasarle como parámetros en la URL el OOID homologado. El OOI se guarda en el *localstorage* del Navegador o en cookies si es posible, entre otros métodos.

El nodo servidor (103), cuyas principales funciones se aprecian en la figura 9.1, dispone de las siguientes funcionalidades:

SETS:
Se puede recibir, almacenar y procesar los eventos de los SDK (104, 105) y nodos (101), utilizándose para ello servidores web, de bases de datos y/o de aplicaciones.

REFS:
Permite enviar y responder a las peticiones de información de los SDKs (104,105) y los Nodos con datos entre otros de SSIDs, *tokens*, perfiles, campañas y resultados, la lista de OUI de fabricantes, un histórico de OUIs disponible antes de que existieran los procesos de aleatorización, y una lista de OUI virtuales o falsas, las lista de OUI validados.

ROGUE:
Puede recibir la lista de SSID a las que se conecta un UT (102) a través del SDK APPs (104). También puede difundir la lista de SSID recibida de los SDK (104,105) a los Nodos (101).

Firmas:
A partir de la combinación de los identificadores de eventos, campañas y respuestas o resultados podría crear una firma OOBS diferente para cada combinación de información enviada al terminal de usuario (102) por mensajes en cualquier lugar o en un lugar determinado y posteriores detecciones realizadas por el Nodo (101); por ejemplo si se detecta en una zona B un terminal de usuario (102) que estaba en una zona A al que dentro del plazo programado de la campaña se le ha invitado por medio de un mensaje a moverse a la zona B.

El objeto de la invención pude ser de interés en aplicaciones de estudios de comportamiento o de análisis de interacción de usuarios. Para ello, el objeto de la invención puede disponer un sistema de análisis de comportamiento como el detallado en la figura 9.2. Dicho sistema

de análisis de comportamiento tiene una serie de funcionalidades que permiten que, a partir de los eventos de Nodo (101) y de los SDK (104,105) en el nodo servidor (103) se calculen o computen visitantes, visitas y visita es [9301], asi como cualquier otro indicador o ratio que se considere de interés a partir de estos para obtener el comportamiento cruzado online y offline de los UTs.

Asimismo, se prevé someter a dichos eventos a los procesos de Homologación, Unificación, Aceptación y Validación y Propagación anteriormente descritos.

Normalmente los nodos (101) generan eventos de tipo Nodo y los SDK (104,105) de tipo SDK; pero los SDK (104,105) podrían generar eventos de tipo nodo, cuando la detección se realiza por *beacon*. Los nodos (101) podrían generan eventos de tipo SDK cuando la detección se realiza al conectarse a un portal cautivo (se genera un evento de tipo SDK) esta información será tratada por sistema de comportamiento como si proviniera del tipo de evento recibido.

Para llevar a cabo un cómputo de visitantes, visitas y visita es en un sitio físico (zona física de alcance del Nodo (101)) o un sitio online (App o Web donde está integrado el SDK APP (104) o en el SDK (105)) se utilizan métodos totalmente análogos. En diferentes implementaciones se pueden implementar modelos similares al mostrado en la figura 9.3. para llegar a obtener los visitantes, o visitas.

Cada vez que se recibe un evento en el Nodo Servidor (103), se registra y se puede analizar teniendo en cuenta la información de eventos anteriores disponibles, durante un período de tiempo. Posteriormente, se pueden obtener diferentes indicadores para diferentes períodos de tiempo como horas, diarios, semanales, mensuales, anuales, etc. Para el análisis de eventos se puede configurar el tiempo de las ventanas de análisis de visitas, y el tiempo del hueco entre las denominadas visita es (concepto análogo a las sesiones en un sitio web, pero aplicable a offline y online). El objetivo de la ventana de análisis es decidir si un evento de tipo Aceptado para el mismo OOID extiende el tiempo de visita u visita de un mismo visitante. El objetivo del hueco entre visita es distinguir diferentes visitas de un mismo visitante considerando que largos períodos de inactividad son debidos a diferentes motivos o intereses en la visita a un espacio físico o digital. En un análisis online el hueco entre visita es, en general es mayor que el hueco entre sesiones, es decir el tiempo de inactividad para considerar que empieza una nueva sesión del mismo visitante. Por lo que el análisis de

eventos comprende configurar: un tiempo de las ventanas de análisis de visitas, donde una ventana de análisis comprende una decisión sobre si un evento de tipo Aceptado para el mismo OOID extiende el tiempo de visita o visita de un mismo visitante, y un tiempo entre visitas, donde dicho tiempo se define como transcurrido entre visitas de un mismo visitante.

Normalmente la ventana de análisis es menor que el hueco y este menor que un día completo. Pasado el tiempo del hueco entre visita es, los eventos forman parte de una nueva visita. Pasado el hueco de visita o sesión, solo los eventos homologados o unificados inician el periodo de la ventana de análisis. Los eventos válidos homologados, unificados o aceptados recibidos dentro del periodo de la ventana de análisis se agruparían en una misma visita, y por tanto se asocian al mismo visitante, y por tanto de la misma visita. Los eventos aceptados permiten extender el tiempo de presencia de una visita del mismo visitante, y por tanto también extender la presencia de una visita.

Con este método las visitas se irán extendiendo dinámicamente a medida que vayan llegando más eventos dentro de unos plazos de tiempo, obteniendo así los tiempos de presencia de una visita en una zona dada, y se podrían realizar análisis del comportamiento online u offline en tiempo real considerando con los datos de eventos recibidos anteriormente, y análisis no en tiempo real con los datos de todo un día o un mes.

A partir de las visitas generadas por un mismo visitante en el módulo de análisis de comportamiento en algunas implementaciones podrían realizarse muchas otras funciones y cómputos, entre otras:

- Podrían establecer criterios para validar detecciones. Entre otros si un mismo visitante esta offline en dos zonas geográficamente muy alejadas.
- Análisis de tiempos medio de presencia.
- Evolución de visitantes.
- Flujos entre varias zonas o sitios online.
- Visitas a un sitio físico que simultánea o previamente usan la App o Web.

Un campo de aplicación de interés específico para la implementación del objeto de la invención es aquel en el que método aquí descrito se complementa con un gestor de perfiles, campañas y resultados como el mostrado en la figura 9.4 donde se aprecia la interacción entre componentes tiene al menos las siguientes funciones que proveen de las siguientes funcionalidades:

Un gestor de perfiles, que tiene entre otras tareas, calcular perfiles o grupos de usuarios que deberán recibir mensajes desde el servidor o por proximidad al ser detectados por el nodo (101), en base a ciertos criterios de segmentación o agrupamiento de comportamiento, o también puede importar/cargar los OOID de los UTs (102) desde una plataforma de análisis de comportamiento externa. A partir de los eventos de Nodo (101) y de los SDK (104,105), de los resultados de visitantes y visitas del sistema de comportamiento, y de las reglas introducidas en el gestor de campañas, el sistema de gestión de perfiles podría calcular o computar qué UTs (102) tienen un mismo perfil de comportamiento cruzado online y offline, así como cualquier otro indicador de *marketing* como un *key performance indicator* (KPI) que se considere de interés a partir de estos datos, un ejemplo de KPI puede ser el número de veces que en media los visitantes con perfil de familia repiten visita en un mes. Por ejemplo, un perfil que podríamos denominar de tipo familia, serían aquellos dispositivos que ayer generaron una visita en la página web de una tienda infantil del centro comercial, hoy han abierto la APP del centro comercial para visitar la cartelera del cine buscando una película infantil y también han sido detectados por un nodo con una presencia de visita de más de 5 minutos en la zona de juegos del centro comercial. Otro ejemplo de definición perfil cruzado online y offline sería aquellos que habitualmente llegan a la oficina con más de 10 minutos antes de la hora de entrada y han usado alguna vez la APP corporativa de la empresa para reservar salas de reuniones. En la definición del perfil pueden incluirse las respuestas a los mensajes recibidos, por ejemplo un perfil de empleados en formación activa serían aquellos empleados que cuando reciben mensajes invitándoles a asistir a cursos presenciales, finalmente acuden al aula donde se realiza dicho curso presencial.

Un sistema de disparo de Mensajes por proximidad física o digital, basado en que cuando un dispositivo es detectado a través de un evento de Nodo (101) además de generar una visita válida y homologada en tiempo real por el sistema de comportamiento, en este momento se comprueba si el OOID está incluido en un perfil activo y una campaña activa, y se lanza de forma inmediata a través del gestor de campañas un mensaje para que le llegue en tiempo real al terminal de usuario (102) por cualquiera de los canales disponibles. Este sistema podría entre otras funciones lanzar los mensajes a los UTs (102) programados en el momento oportuno según la información programada en el gestor de campañas al recibir eventos de Nodo (101), de Nodo Virtual (103) o de los SDK (104,105) entre otros eventos disparadores. Es decir, por proximidad o visita a un sitio físico o un sitio digital.

Se tienen como canales online disponibles todos aquellos que permiten hacer un envío directo al terminal de usuario (102) sin necesidad de que el terminal de usuario (102) solicite previamente el mensaje (es decir, tipo *push*). Para ello se dispone de identificadores para la comunicación (*tokens*) de cada terminal de usuario (102). Entre otros los canales se eligen de entre: Notificaciones *Push* APP, *Email*, SMS, Notificaciones *Push Web*] o Notificaciones *push* de navegador. Para el envío de mensajes a los UTs (102), se pueden ligar los OOID Homologados o no Homologados a cada entidad de Usuario (es decir la entidad que almacene dichos *tokens* de comunicación y otros datos personales) en el Nodo servidor (103). Aunque un terminal de usuario (102) no tenga OOID homologado puede recibir mensajes desde el Nodo servidor (103), pero necesitaría el OOID homologado y validado para hacer el envío *push* de mensajes por proximidad al nodo (101).

El gestor de campañas entre otras tareas podría generar o cargar previamente a su comunicación los contenidos y formatos de los mensajes para los diferentes canales elegidos, así como fechas, zonas, sitios online, y cualquier otra variable útil para la definición de la campaña. Se prevé, lanzar las campañas y mensajes en el momento programado en el gestor de campañas los mensajes a los UTs (102) programados. Es decir, por tiempo y perfil; también se prevé importar campañas y mensajes de terceros mediante *webservices* u otros procedimientos, para completar la funcionalidad del gestor de campañas.

El gestor de resultados, entre otras tareas, recoge la información de confirmación en envío de mensajes de los *proxies* de cada canal, las respuestas asociadas a los espacios digitales que se han utilizado en la comunicación (rellenado de formulario datos personales, encuestas, subscripciones, etc..), así como su comportamiento en el espacio físico y digital tras el envío de cupones, ofertas, información, encuestas, otras formas de sugerir visitar sitios físicos o digitales, etc... Por tanto, sería capaz de calcular por ejemplo cuantos usuarios que han recibido una oferta por email el viernes en la tarde, han sido detectados en el supermercado durante el sábado por la mañana.

En función de la información recibida del resultado del mensaje, se puede actualizar la información del gestor de campañas. El gestor de resultados además de reportar datos de eficacia, puede actuar de lazo de realimentación para modificar en tiempo real los perfiles y campañas, consiguiendo una optimización dinámica en función de la respuesta del usuario

Gracias a las funcionalidades anteriores el sistema posibilita realizar lo que se conoce como *remarketing* pero de forma offline, que consiste en enviar un mensaje al terminal de usuario (102) por proximidad o presencia offline, conociendo previamente sus gustos o intereses a través del comportamiento digital analizado, por ejemplo en las APPs o sitios WEB que tienen el SDK integrado, o a través del comportamiento físico analizado en las zonas donde hay nodos, e incluso basado en respuestas a comunicaciones enviadas anteriormente.

REIVINDICACIONES

1. Método de identificación única de equipos de usuario (102) que son capaces de ejecutar aplicaciones móviles que integran un SDK APP (104), y un navegador web que a su vez integra un SDK web (105); caracterizado porque utiliza:

- una serie de nodos interconectados entre sí (101) que son dispositivos con capacidad de proceso equipados con medios para escuchar y procesar información de señales radiofrecuencia emitidas por el equipo de usuario (102),
- un nodo servidor (103) configurado para analizar y propagar identificadores de equipo de usuario (102),
- caracterizado porque la identificación se consigue cruzando datos análisis en tiempo real y de históricos de eventos generados por el equipo de usuario (102) y obtenidos por SDK WEB (105), al usar APPs con el SDK APP (104), y los datos de las tramas recibidas por los dispositivos físicos en zonas físicas dentro de una zona de cobertura de los nodos (101).

Dando como resultado la generación de al menos un identificador único, (OOID).

Y caracterizado por que el identificador se actualiza con nuevos datos (firmas), que incrementan su certidumbre, a partir de una combinación de firmas que se van añadiendo de forma incremental y por que una vez obtenido el identificador, este se difunde entre el servidor (103) los nodos (101) y los SDK (104 y 105).

2. Método según la reivindicación 1 caracterizado porque adicionalmente comprende implementar un sistema de disparo de Mensajes por proximidad física o digital, que comprende detectar:

a. un dispositivo es detectado a través de un evento de Nodo (101),
b. generar una visita válida y homologada en tiempo real,
c. comprobar si el OOID está incluido en al menos uno de: un perfil activo y una campaña activa, y
d. lanzar un mensaje para que le llegue en tiempo real equipo de usuario (102) por cualquiera de los canales disponibles.

3. Método según una cualquiera de las reivindicaciones anteriores caracterizado porque adicionalmente comprende implementar privacidad en la localización e identificación frente a terceros de los equipos de usuario (102) que están en el espacio físico de alcance del Nodo

(101) , donde el nodo (101) genera en tiempo real o de ejecución tramas de descubrimiento de las redes WLAN o BT, por los equipos de usuario (102), usando para ello:

a. las mismas direcciones de la capa de acceso al medio de los equipos de usuario (102)) escuchados recientemente, o

b. las mismas direcciones de la capa de acceso al medio de los equipos de usuario (102)) escuchados por otros nodos (101)

que han sido recibidas por el proceso de propagación, y también generarlas aleatoriamente con diferentes patrones comunes similares a los recibidos.

4. Método según una cualquiera de las reivindicaciones anteriores caracterizado porque adicionalmente comprende calcular un cómputo de visitantes y visitas donde dicho cálculo comprende a su vez:

a. recibir y registrar un evento del nodo servidor (103) cada vez que se recibe,

b. analizar dicho evento teniendo en cuenta la información de eventos anteriores disponibles, durante un período de tiempo,

c. obtener diferentes indicadores para diferentes períodos de tiempo

5. Método según reivindicación 4 donde el análisis de eventos comprende configurar:

a. tiempo de las ventanas de análisis de visitas, donde una ventana de análisis comprende una decisión sobre si un evento de tipo Aceptado para el mismo OOID extiende el tiempo de visita o visita de un mismo visitante, y

c.. tiempo entre visitas, donde dicho tiempo se define como transcurrido entre visitas de un mismo visitante.

6. Método según reivindicación 1 4 caracterizado por que comprende gestionar la propagación del identificador (OOID) de los equipos de usuario (102) haciendo disponible dicho identificador (OOID) al usar el equipo de usuario (102) para navegar a páginas web mediante el SDK WEB (105), al usar APPs con el SDK APP (104) y al entrar en zonas físicas dentro de una zona de cobertura de los nodos (101).

7. Método según reivindicación 1 caracterizado por que comprende homologar el identificador (OOID) del SDK APP (104) o el identificador (OOID) del SDK WEB (105) en aquellos equipos de usuario (102) que no permiten acceder a la dirección de la capa de acceso al medio por software.

8. Método según una cualquiera de las reivindicaciones anteriores caracterizado porque adicionalmente comprende calcular un cómputo de visitantes tanto offline, provenientes de la detección en el espacio físico dentro del área de cobertura del Nodo (101), como online, visitas a través de la navegación a webs y uso de apps, donde dicho cálculo comprende a su vez:

a. recibir y registrar un evento del nodo servidor (103) cada vez que se recibe

b. analizar dicho evento teniendo en cuenta la información de eventos anteriores disponibles, durante un período de tiempo, y

c. obtener diferentes indicadores para diferentes períodos de tiempo

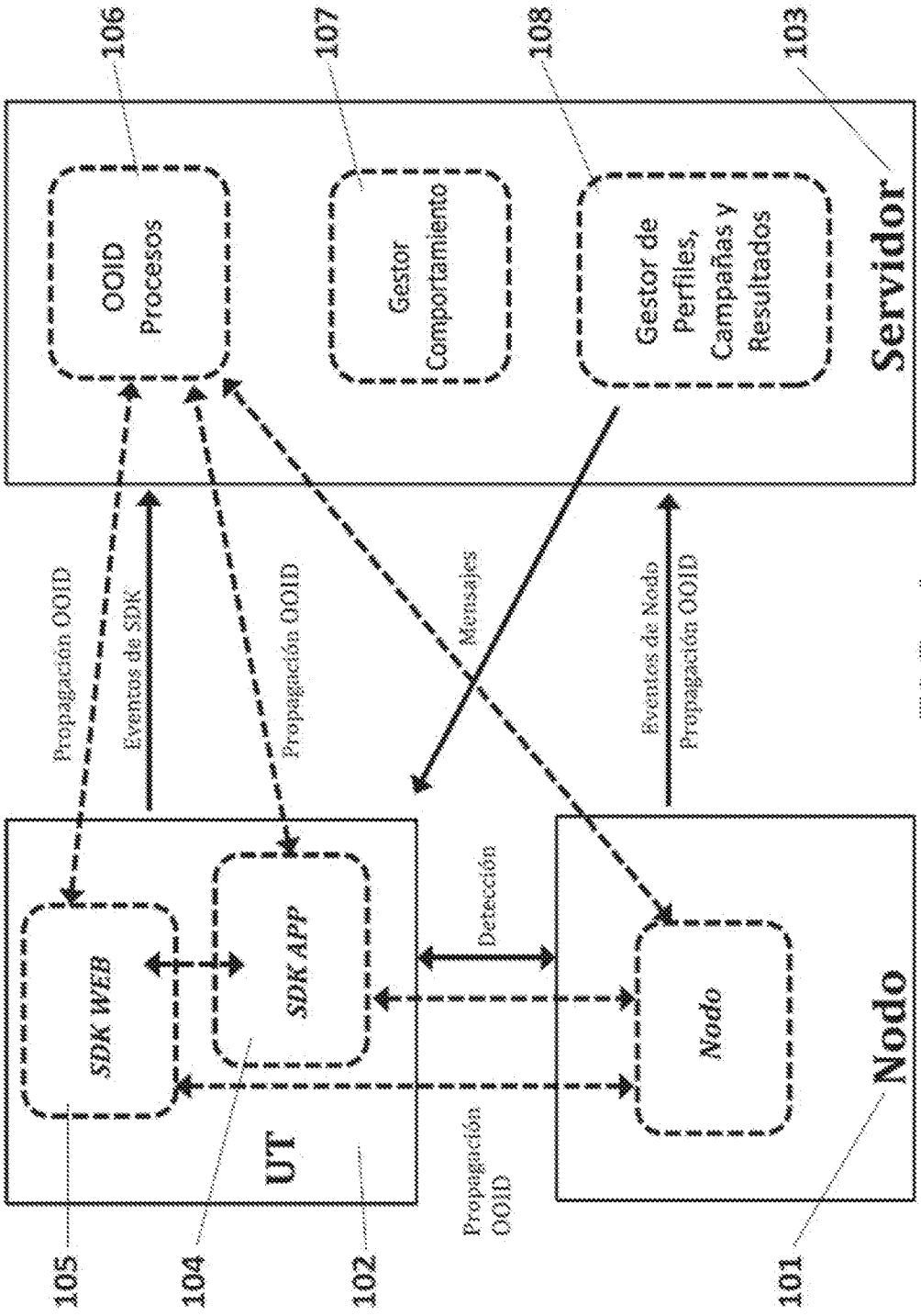
106
107
108
103
OOID Procesos
Gestor Comportamiento
Gestor de Perfiles, Campañas y Resultados
Servidor
Propagación OOID
Eventos de SDK
Propagación OOID
Mensajes
Eventos de Nodo
Propagación OOID
SDK WEB
SDK APP
Detección
Nodo
UT
Propagación OOID
105
104
102
101

FIG. 1

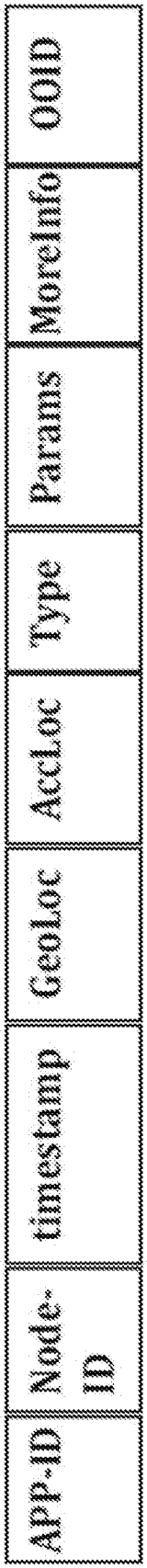
APP-ID
Node-ID
timestamp
GeoLoc
AccLoc
Type
Params
MoreInfo
OOID

FIG. 2

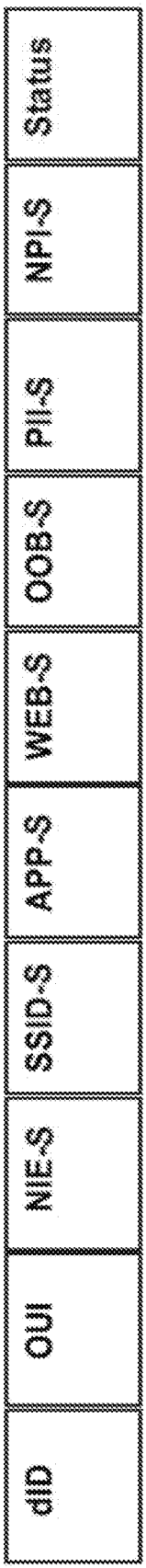
dID
OUI
NIE-S
SSID-S
APP-S
WEB-S
OOB-S
PII-S
NPI-S
Status

FIG. 3

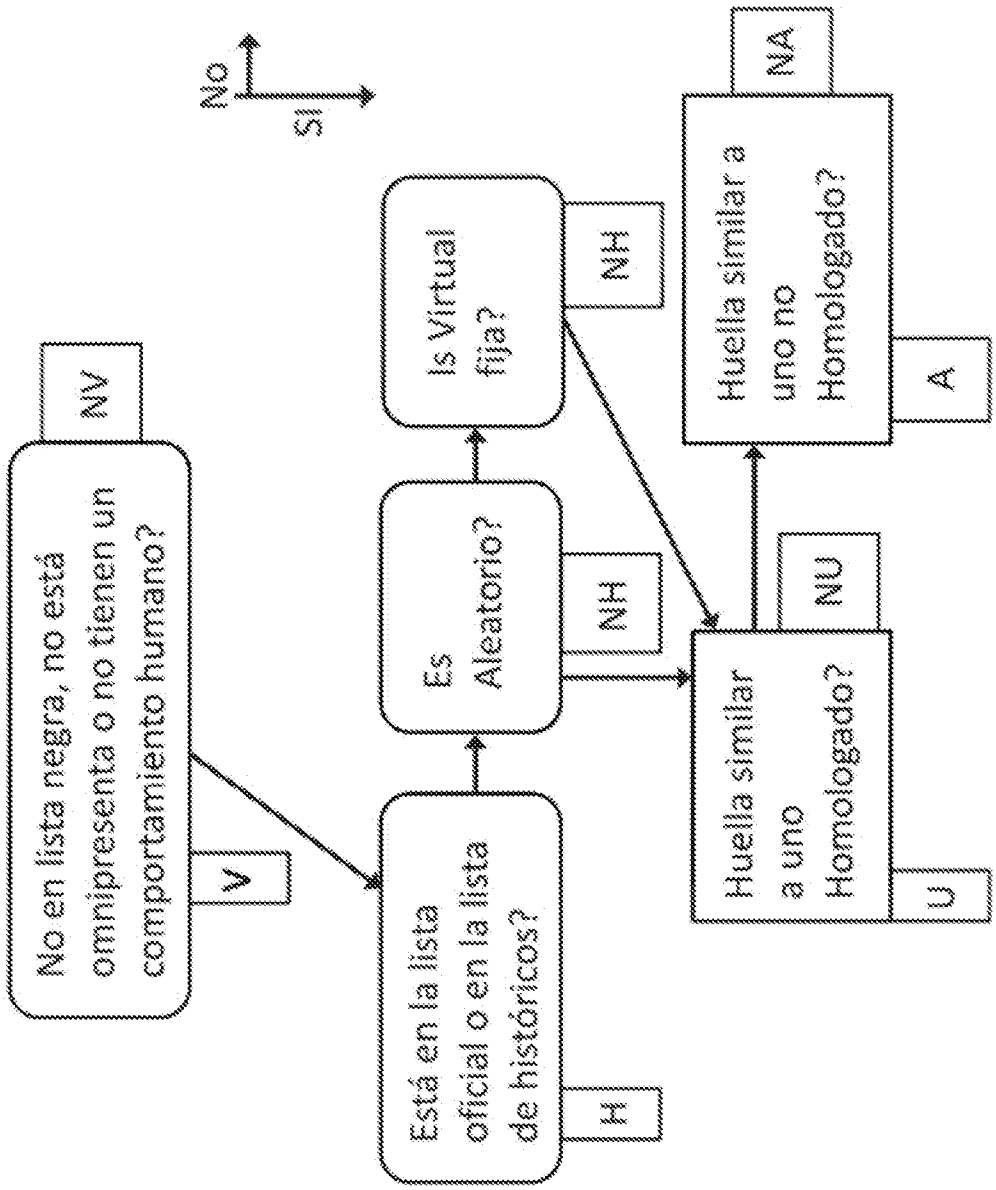
No en lista negra, no está omnipresenta o no tienen un comportamiento humano?
V
NV
Está en la lista oficial o en la lista de históricos?
H
Es Aleatorio?
NH
Is Virtual fija?
NH
Huella similar a uno Homologado?
U
NU
Huella similar a uno no Homologado?
A
NA
No
SI

FIG. 4

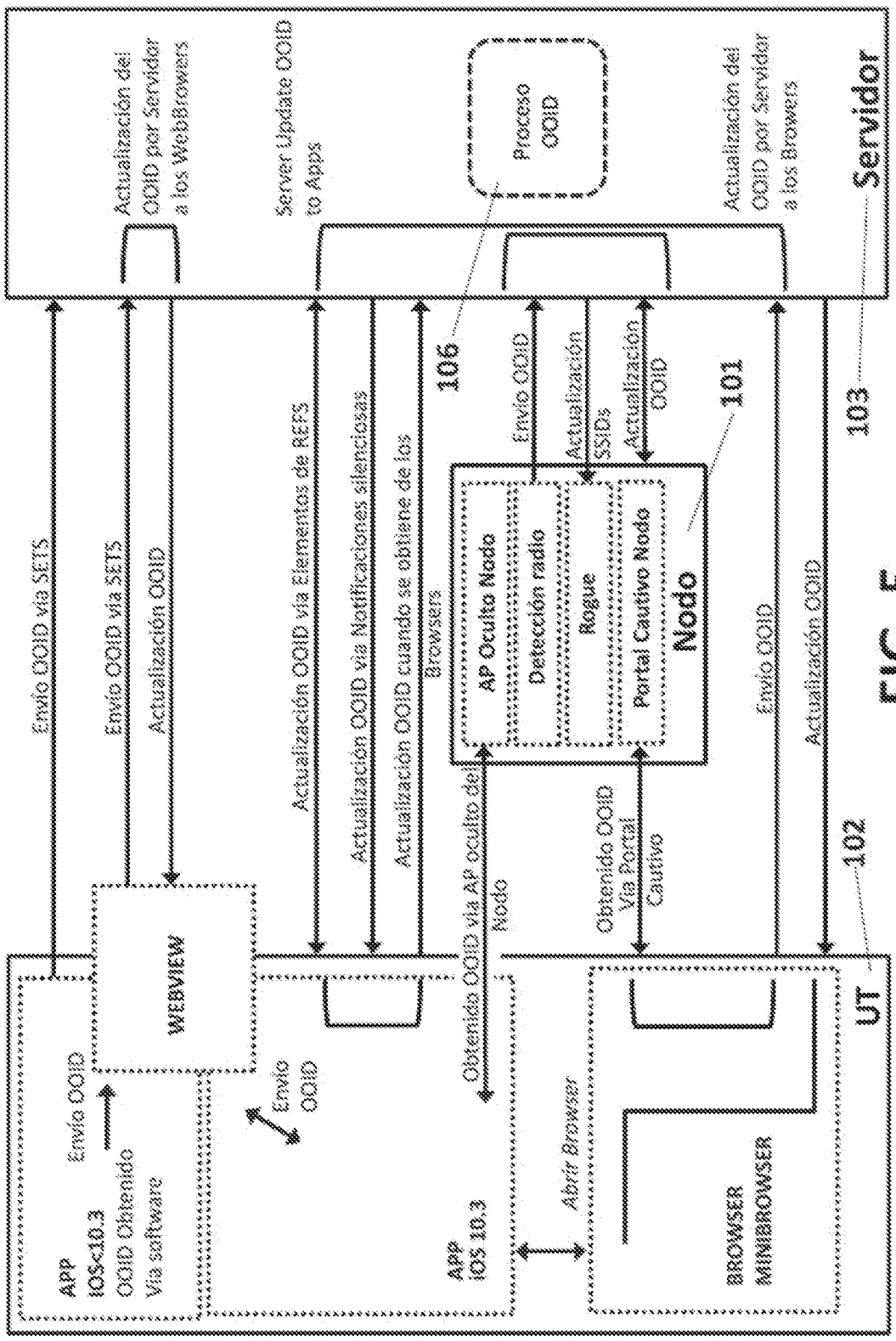
Servidor
Actualización del OOID por Servidor a los WebBrowsers
Server Update OOID to Apps
Proceso OOID
Actualización del OOID por Servidor a los Browsers
Envío OOID via SETS
Envío OOID via SETS
Actualización OOID
Actualización OOID via Elementos de REFS
Actualización OOID via Notificaciones silenciosas
Actualización OOID cuando se obtiene de los Browsers
106
Envío OOID
Actualización SSIDs
Actualización OOID
101
103
AP Oculto Nodo
Detección radio
Rogue
Portal Cautivo Nodo
Nodo
Envío OOID
Actualización OOID
Obtenido OOID via AP oculto del Nodo
Obtenido OOID Via Portal Cautivo
102
APP IOS<10.3 OOID Obtenido Via software
Envío OOID
WEBVIEW
Envío OOID
APP IOS 10.3
Abrir Browser
BROWSER MINIBROWSER
UT

FIG. 5

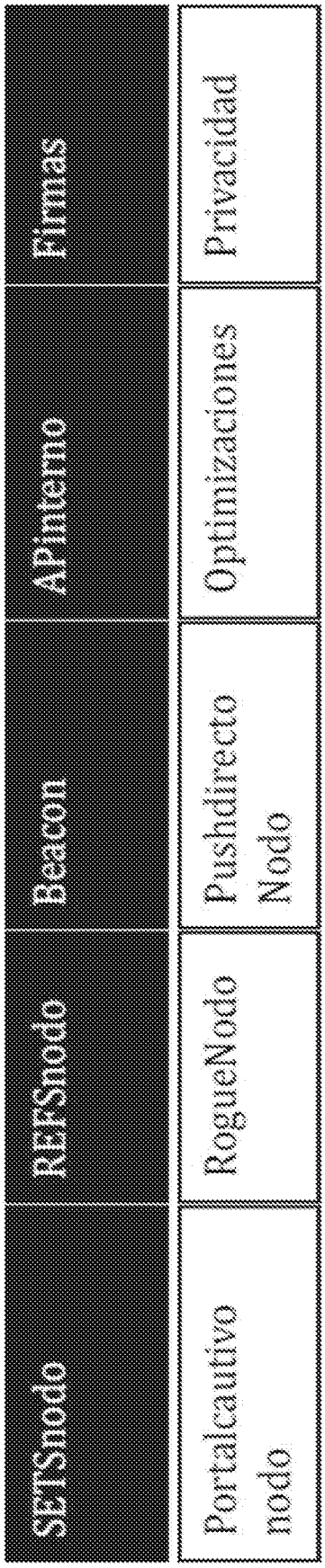
SETSnodo
REFSnodo
Beacon
APinterno
Firmas
Portalcautivo nodo
RogueNodo
Pushdirecto Nodo
Optimizaciones
Privacidad

FIG. 6.1

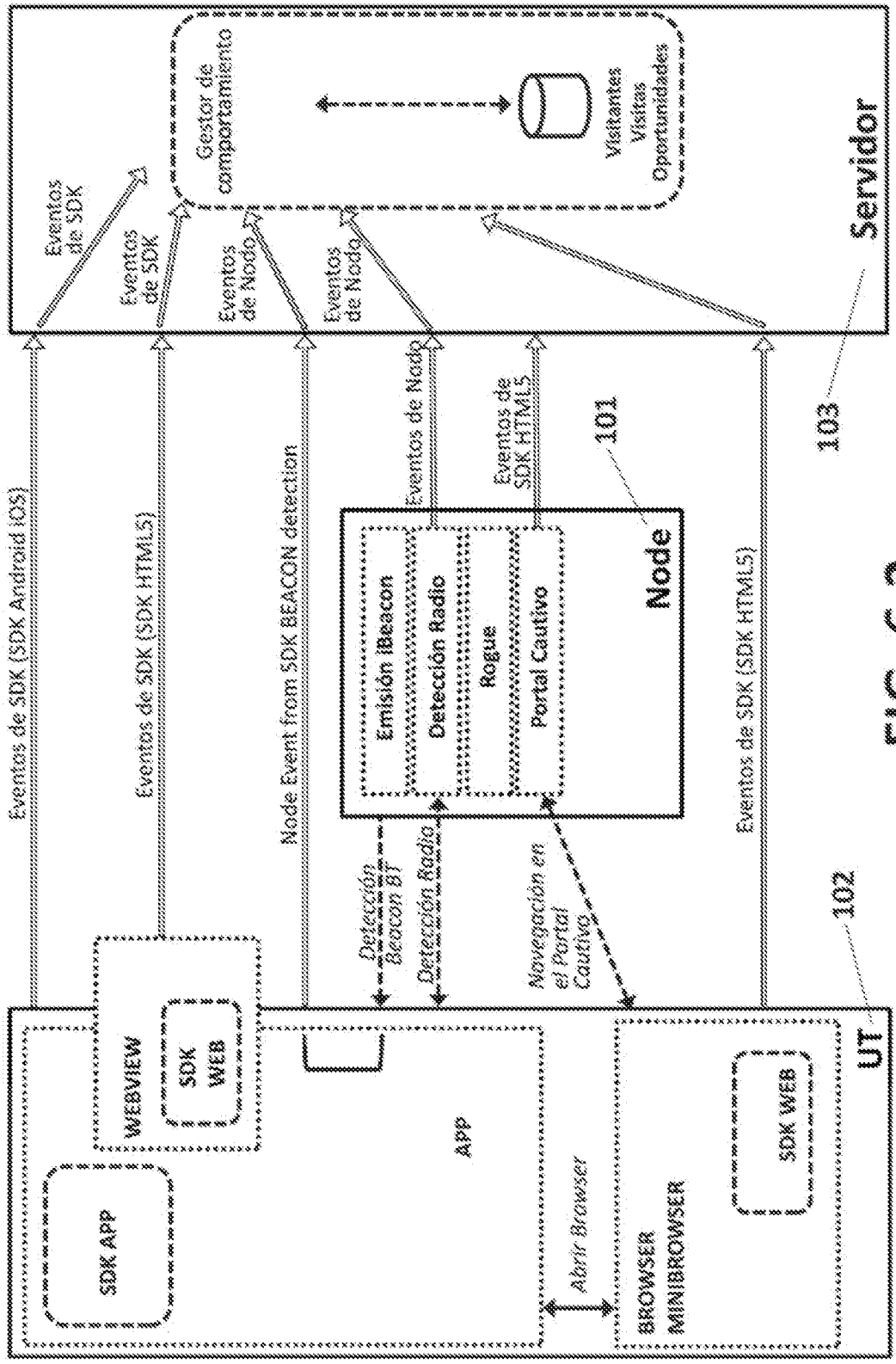
Servidor
Gestor de comportamiento
Visitantes
Visitas
Oportunidades
Eventos de SDK
Eventos de SDK
Eventos de Nodo
Eventos de Nodo
Eventos de SDK (SDK Android iOS)
Eventos de SDK (SDK HTML5)
Node Event from SDK BEACON detection
Eventos de Nodo
Eventos de SDK HTML5
101
103
Node
Emisión iBeacon
Detección Radio
Rogue
Portal Cautivo
Eventos de SDK (SDK HTML5)
Detección Beacon BT
Detección Radio
Navegación en el Portal Cautivo
102
UT
SDK APP
WEBVIEW
SDK WEB
APP
Abrir Browser
BROWSER MINIBROWSER
SDK WEB

FIG. 6.2

SETSSDK	RERSSDK	Seguimiento	AP interno	Firmas
Geofence	RogueSDK	Pushdirecto SDKAPP	Tokenydatos personales	Privacidad
Interrogation	Propagación			

FIG. 7

SETSSDK	RERSSDK	Seguimiento s	Notificaciones	Firmas
	Propagación	Pushdirecto SDKWeb	Tokenydatos personales	Privacidad

FIG. 8

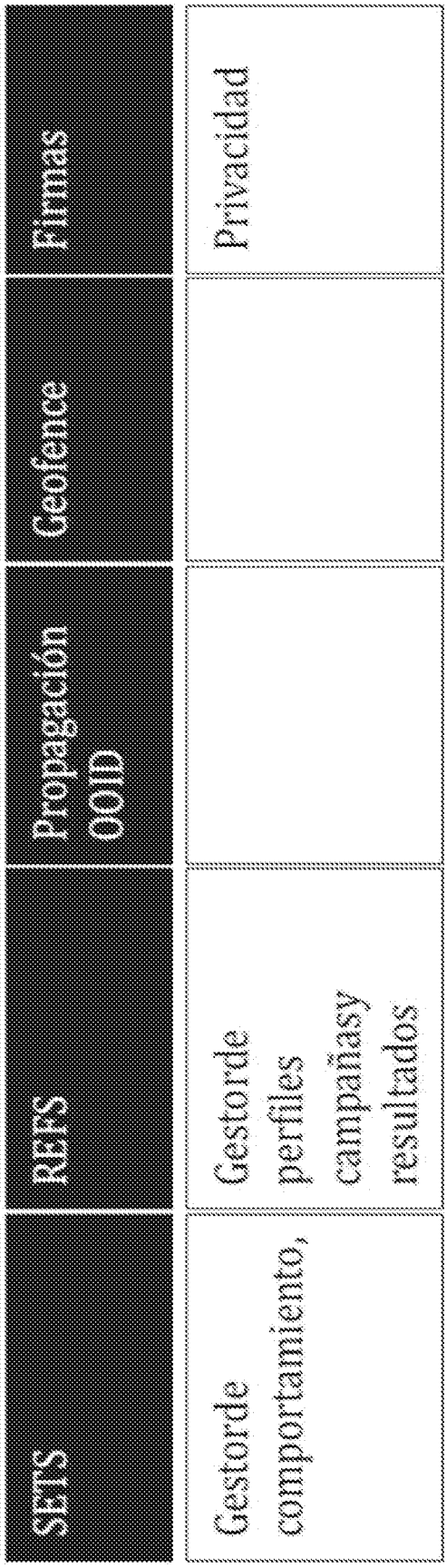
SETS
REPS
Propagación OOID
Geofence
Firmas
Gestor de comportamiento,
Gestor de perfiles campañas y resultados
Privacidad

FIG. 9.1

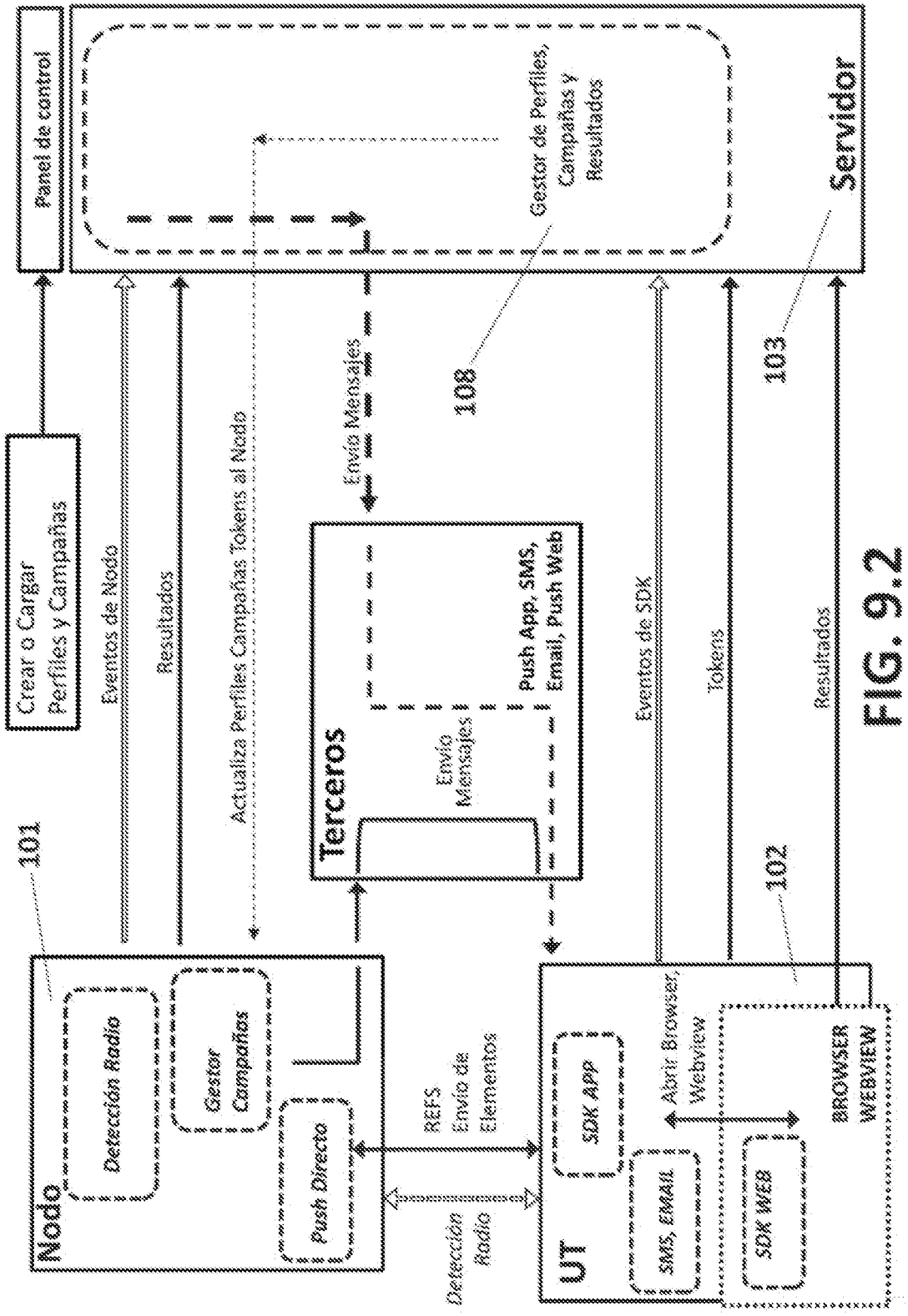
Panel de control
Servidor
Gestor de Perfiles, Campañas y Resultados
Crear o Cargar Perfiles y Campañas
Eventos de Nodo
Resultados
Actualiza Perfiles Campañas Tokens al Nodo
Envío Mensajes
108
103
Terceros
Envío Mensajes
Push App, SMS, Email, Push Web
101
Nodo
Detección Radio
Gestor Campañas
Push Directo
REFS Envío de Elementos
Detección Radio
UT
SDK APP
SMS, EMAIL
SDK WEB
Abrir Browser, Webview
BROWSER WEBVIEW
Eventos de SDK
Tokens
Resultados
102

FIG. 9.2

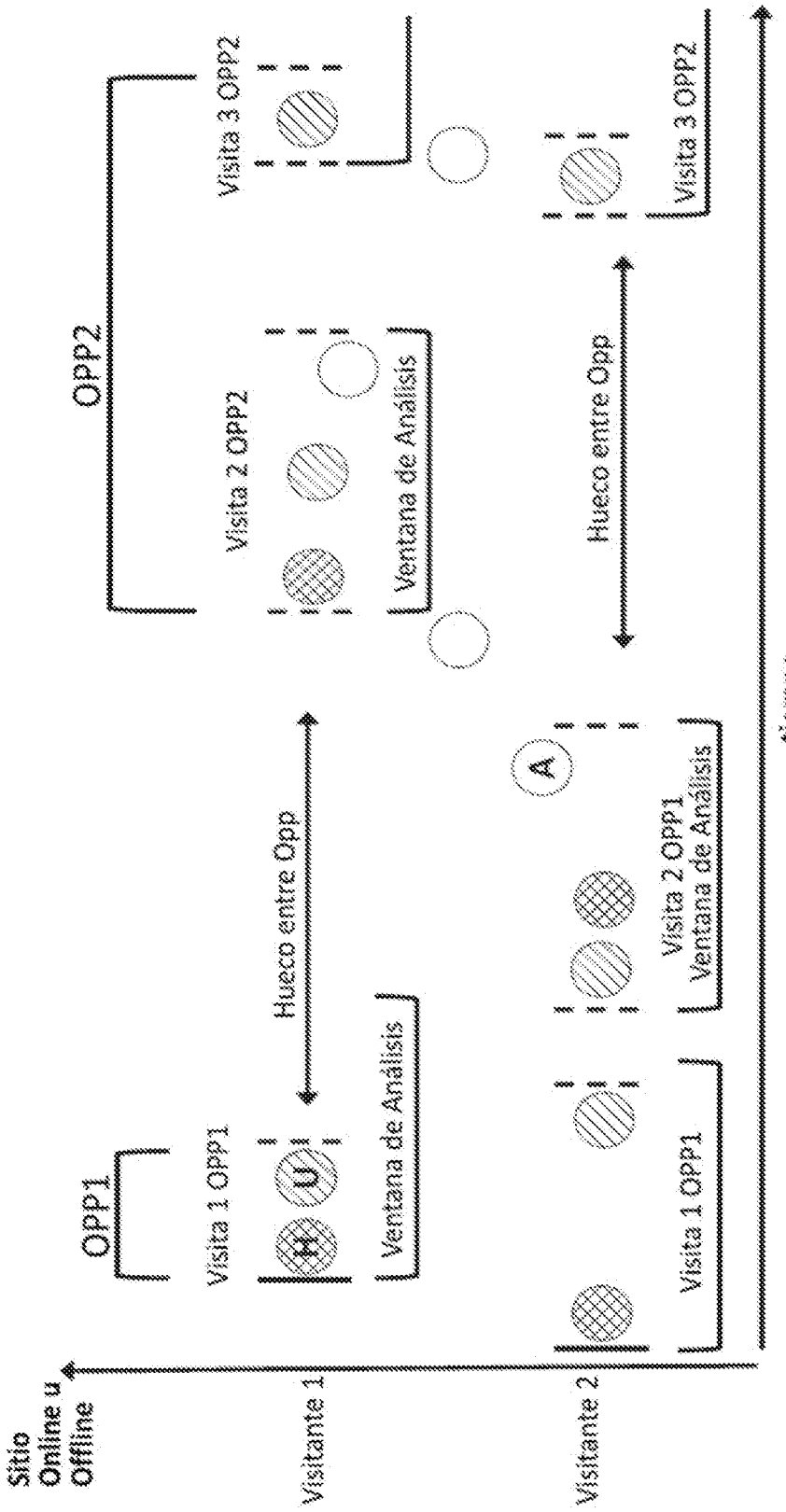
Sitio
Online u
Offline
Visitante 1
Visitante 2
OPP1
Visita 1 OPP1
H
U
Hueco entre Opp
Ventana de Análisis
A
Visita 1 OPP1
Visita 2 OPP1
Ventana de Análisis
OPP2
Visita 2 OPP2
Ventana de Análisis
Hueco entre Opp
Visita 3 OPP2
Visita 3 OPP2
tiempo

FIG. 9.3

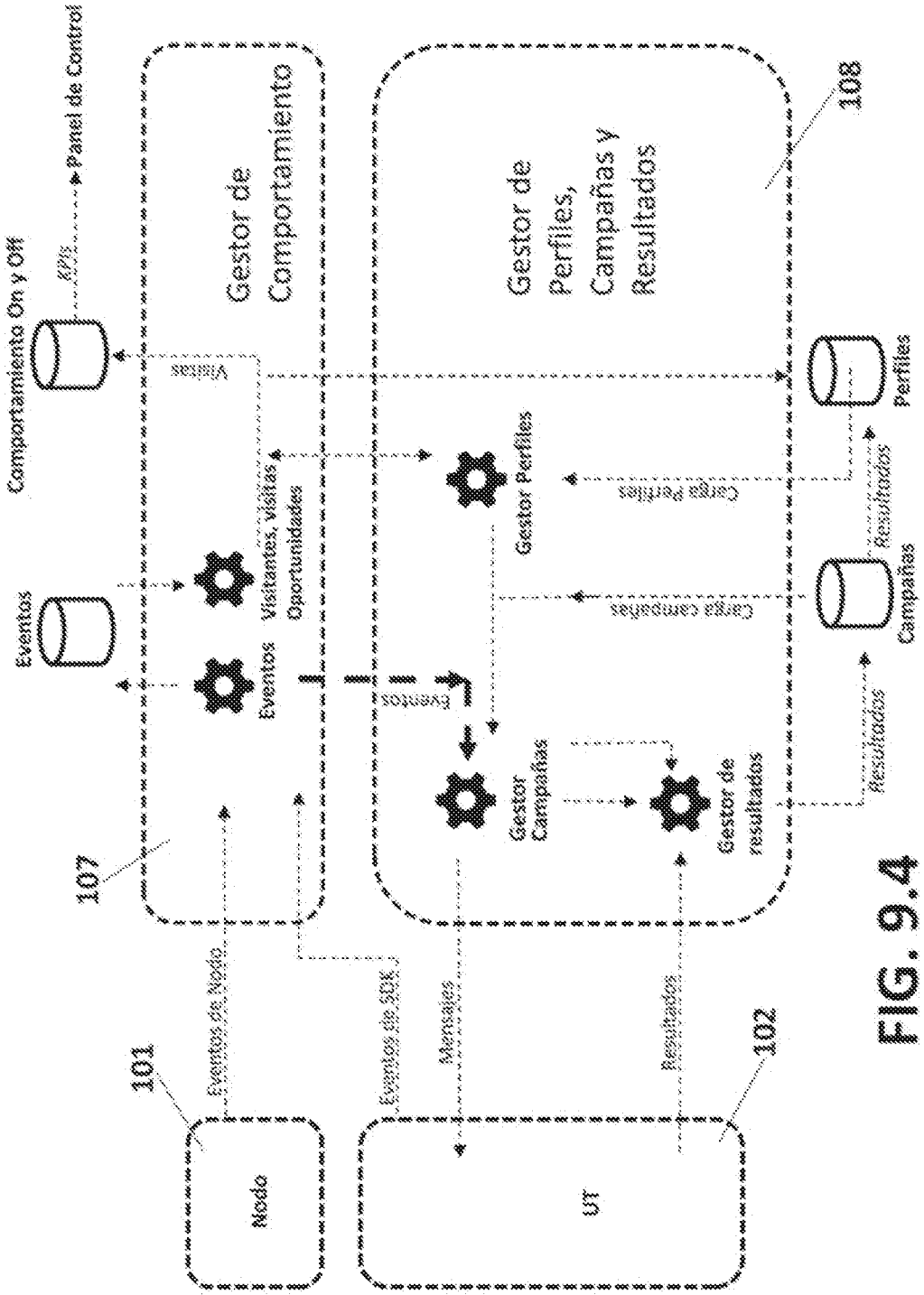
Comportamiento On y Off
KPIs
Panel de Control
Gestor de Comportamiento
Gestor de Perfiles, Campañas y Resultados
108
Eventos
Visitantes, visitas
Oportunidades
Eventos
Gestor Perfiles
Carga Perfiles
Perfiles
Resultados
Carga campañas
Campañas
Gestor Campañas
Gestor de resultados
Resultados
107
Eventos de Nodo
Eventos de SDK
Mensajes
Resultados
101
102
Nodo
UT

FIG. 9.4