Universidad
Rey Juan Carlos

TESIS DOCTORAL

Gestión de identidades y accesos
en el Internet de las Cosas

Autor:
Elías Grande Rubio

Directora:
Dra. Marta Beltrán Pardo

Programa de Doctorado en Tecnologías de la Información y las Comunicaciones

Escuela Internacional de Doctorado

2021

DEPARTAMENTO DE CIENCIAS DE LA
COMPUTACIÓN, ARQUITECTURA DE
COMPUTADORES, LENGUAJES Y SISTEMAS
INFORMÁTICOS Y ESTADÍSTICA E
INVESTIGACIÓN OPERATIVA

Gestión de identidades y accesos en el Internet de las Cosas

TESIS DOCTORAL

Autor: Elías Grande Rubio
Ingeniero Superior en Informática

Directora: Dra. Marta Beltrán Pardo
Doctora en Informática

2021

Gran parte de las dificultades por las
que atraviesa el mundo se deben a que
los ignorantes están completamente seguros
y los inteligentes llenos de dudas.
Bertrand Russell

No rompas el silencio
si no es para mejorarlo.
Ludwing Van Beethoven

Agradecimientos

En primer lugar, he de agradecer a Marta toda su paciencia y dedicación conmigo a lo largo de todo el desarrollo de esta tesis. Trabajar a tiempo completo a la vez que se lleva a cabo los estudios doctorales puede ser algo agotador y provocar que se pierda el foco del objetivo debido a las múltiples fuentes de distracción. Aún con todo esto, siempre ha estado ahí como fuente de ánimo y de inspiración.

En segundo lugar, no creo que hubiera sido capaz de llegar hasta aquí sin el pilar fundamental de mi familia. La familia es algo que en muchas ocasiones puede parecer transparente en el camino pero sin embargo, para mí ha sido tan necesaria, que gracias a su apoyo y motivación me ha permitido no tirar la toalla en ninguna de las empresas que he emprendido hasta la fecha, y espero que siga siendo así.

Finalmente, agradecer a todo el conjunto de personas que de una forma u otra, y posiblemente muchas veces sin saberlo, han ido dibujando las diferentes líneas de mi personalidad tanto a nivel personal como profesionalmente de tal forma que me han convertido en la persona que soy a día de hoy. No todo se aprende de los libros, y este conjunto de personas de los que no daré nombres para no olvidarme de ninguno, son prueba de ello.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. La gestión de identidades y accesos

De la misma forma que sucede en el mundo físico cuando las diferentes autoridades competentes solicitan el documento nacional de identidad a cada individuo para identificarlo, en el mundo digital también se necesita conocer quién está accediendo en cada momento a uno u otro recurso y si dicho usuario tiene los permisos necesarios para realizar o no dicha tarea. Dentro del mundo digital, la disciplina que se encarga de abordar esta problemática se denomina gestión de identidades y accesos.

Desde el momento en el que una persona física o jurídica inicia su relación con un sistema digital, es necesario crear su identidad digital en base a la traducción total o parcial de su identidad en el mundo físico. Dicha identidad digital posee un ciclo de vida en el puede ser dotada, entre otros, de diferentes atributos que enriquezcan o añadan información a dicha identidad, de un conjunto de permisos de acceso recursos protegidos o ejecución de tareas, y de una gestión de métodos de autenticación soportados por el sistema para probar la identidad del usuario que hay detrás cuando dicho sistema lo requiera.

En la mayor parte de los casos, la gestión de identidades y accesos pivota sobre uno de sus pilares fundamentales: la autenticación, la cual no es más que un conjunto de retos que el sistema realiza al usuario en base a algo que conoce (una contraseña), algo que tiene (un teléfono móvil), o algo que es (datos biométricos),

con el fin de validar que el usuario es quien dice ser y no está intentando suplantar a un tercero. De esta forma, una vez autenticado el usuario, éste puede disfrutar de los diferentes permisos vinculados a su identidad digital.

Una vez se ha identificado al usuario dentro del sistema y se ha llevado a cabo su autenticación, el comportamiento de dicho usuario es controlado gracias a la gestión de la autorización en el control de acceso a los diferentes recursos. Además, la gestión de auditoría por parte del sistema se encarga de registrar cada una de las diferentes acciones que realiza cada usuario, ya sean acciones permitidas o no en función de sus permisos, con fines forenses si en el futuro fuera necesario llevar a cabo una investigación sobre cualquier comportamiento anómalo detectado en el sistema.

Todas estas capacidades, Identificación, Autenticación, Autorización y Auditoría, son las que dentro del ámbito de la gestión de identidades y accesos, conforman el acrónimo IAAA. Dicho acrónimo y las capacidades que representa se encarga, en resumen, de garantizar los niveles de confidencialidad, disponibilidad, integridad y no repudio del sistema en el cual se está materializando dicha gestión.

1.2. Internet de las cosas

La tendencia de crecimiento del número de dispositivos conectados a Internet en los últimos años ha sido exponencial y parece que continuará así durante los próximos años. Ya no son ordenadores o teléfonos móviles los únicos dispositivos conectados sino que cada vez más, encontramos diferentes tipos de sensores y actuadores embebidos en el mundo físico que nos rodea. Algunos ejemplos cercanos de dónde se encuentran este tipo de dispositivos pueden ser frigoríficos, lavadoras, bombillas, vehículos y un largo etcétera, que dan vida al gran entramado de dispositivos conectados a nuestro alrededor. Estos pequeños dispositivos que la mayor parte de las veces pasan desapercibidos en nuestra vida cotidiana forman parte del denominado Internet de las cosas (en ingles, *Internet of Things* - IoT). Este concepto del Internet de las cosas agrupa bajo su paraguas un enorme ecosistema de tecnologías y protocolos de comunicación con ciertas características que facilitan la implantación de este tipo de dispositivos con el fin de cumplir la función concreta que se necesite dentro del dominio de aplicación en el que se

despliegen.

El bajo coste de estos dispositivos ha provocado que diferentes proveedores, investigadores y personas interesadas en este campo se dispongan a explorar las múltiples casuísticas en los que pueden ser aplicados, confiriéndole a dicho campo, un nivel de heterogeneidad sin precedentes debido a la gran diversidad de componentes y tecnologías que pueden ser utilizadas en la fabricación de los dispositivos. Además, su facilidad para ser construidos y desplegados dotan a cualquier dominio de aplicación de niveles de escalabilidad similares a los entornos automatizados *cloud*.

Sin embargo, a diferencia de los servicios *cloud* desplegados en servidores con recursos casi ilimitados, los dispositivos englobados dentro del Internet de las cosas plantean multitud de retos al no disponer de las mismas capacidades de cómputo, almacenamiento seguro, soporte de protocolos de comunicación robustos o incluso en el uso de energía, lo que convierte a este ecosistema en algo así como el hermano pequeño del *cloud* desde el punto de vista de recursos disponibles.

La similitud desde el punto de vista de la escalabilidad con el *cloud* y la facilidad a la hora de construir los dispositivos, está provocando que se utilicen paradigmas de computación diseñados principalmente para el *cloud* en entornos que no soportan los mismos niveles de confiabilidad y seguridad, todo ello en aras de un mejor posicionamiento en el mercado antes que el resto de los competidores comerciales. Por este motivo, muchos de los problemas identificados en los despliegues de soluciones del Internet de las cosas y listados en el *OWASP IoT Top 10* [1] como vulnerabilidades reiteradas, están relacionadas en su mayoría con las propias características intrínsecas de recursos limitados ya conocidas de dichos dispositivos como son las limitaciones de cómputo que les impide utilizar cifrados robustos, la necesidad de utilizar protocolos de comunicación ligeros que no soportan características de seguridad o la falta de almacenamiento seguro, entre otros.

Aún con todos estos retos presentes, la infinidad de beneficios que aportan gracias a la información que recaban del entorno en el que están desplegados promueve que el estudio de su despliegue abarque múltiples dominios de aplicación como domótica, sanidad, urbanismo, agricultura y ganadería, control medioam-

biental, vehículos autónomos y transporte, control de infraestructuras y edificios, turismo y actividades deportivas entre otras, lo cual es la principal motivación que impulsa este tipo de tecnología y será por tanto, lo que mantendrá su crecimiento exponencial en los años venideros.

1.3. Tema de investigación

Como ya se ha comentado en la sección anterior, la incorporación en nuestro día a día de dispositivos englobados dentro del Internet de las cosas es inevitable debido a la gran cantidad de oportunidades y beneficios que puede ofrecer en diferentes campos que mejoran nuestra calidad de vida y convierten a nuestros entornos en inteligentes. Sin embargo, el entorno altamente competitivo en el que se está moviendo dicha industria está empujando a realizar diseños que cubren sólo las necesidades funcionales básicas de los diferentes dominios de aplicación sin poner foco en aspectos como la seguridad o la privacidad, lo que puede provocar diferentes tipos de incidentes con el correspondiente coste económico y reputacional que eso conlleva.

Las características intrínsecas de los propios dispositivos dificultan su integración de manera segura en los diferentes sistemas de gestión de identidades y accesos ya sea, por sus limitaciones en capacidades de almacenamiento seguro de las credenciales necesarias para llevar a cabo los diferentes procesos de autenticación requeridos por el sistema, o por no soportar protocolos de comunicación robustos desde el punto de vista de seguridad. Esto provoca dificultades cuando es necesario confiar en la identidad unívoca de un dispositivo dentro de un contexto IoT, no hay un nivel de seguridad suficiente para saber si ha sido o no suplantada. Además, los problemas de escalabilidad que plantea la gran cantidad de esta clase de dispositivos englobados dentro de un dominio de aplicación concreto termina forzando a los diferentes fabricantes a utilizar configuraciones y credenciales por defecto a la hora de construir en masa los diferentes dispositivos, empeorando dicha problemática. Finalmente, este conjunto de dificultades de identificar un dispositivo y confiar en que es quien dice ser, derivan en limitaciones de comunicación por parte de los servicios *cloud* hacía dichos dispositivos al no disponer de mecanismos de direccionamiento o nombrado estándares, efectivos y robustos.

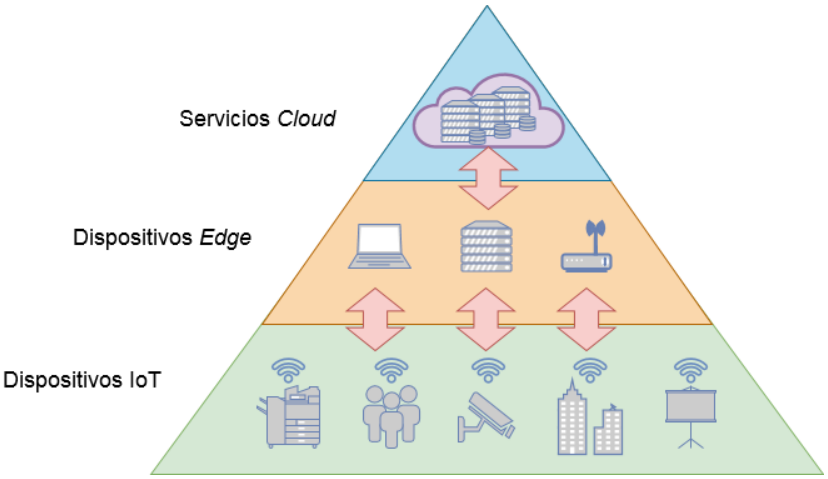

Figura 1.1: Arquitectura de tres capas con enfoque *edge-centric* para IoT

Y hay que tener en cuenta que esta comunicación entre servicios desplegados en centros de datos remotos y los dispositivos empotrados en la realidad física, es la base de la mayor parte de proyectos IoT en la actualidad. En muchos de estos proyectos se busca la obtención de un conocimiento colectivo construido a partir de los datos captados por los sensores desplegados en la realidad física, que permita tomar decisiones que redunden positivamente en la calidad de los servicios proporcionados a los usuarios o ciudadanos. Pero de nuevo, sin los mecanismos de seguridad y privacidad adecuados, estas aplicaciones del conocido como *crowd-sensing* pueden implicar riesgos para la seguridad y la privacidad.

Debido a este conjunto de dificultades, los diferentes trabajos de investigación y proyectos llevados a cabo por la industria intentan cubrir dichas deficiencias mediante dos enfoques. El primero, y posiblemente menos acertado, es el de incorporar la tecnología necesaria para cada caso en los diferentes dispositivos, lo que deriva en un encarecimiento de los costes de producción y en un incremento en el uso de la energía disponible en las baterías de los dispositivos al no haber sido diseñadas inicialmente para dicha funcionalidad extra. El segundo enfoque, que está tomando bastante fuerza, es el de incluir una capa de dispositivos intermedios entre los servicios *cloud* y los dispositivos embebidos en el mundo físico que abstraen a estos últimos de los protocolos de comunicación necesarios para comunicarse con el *cloud*, mejorando las latencias del sistema y haciendo un uso

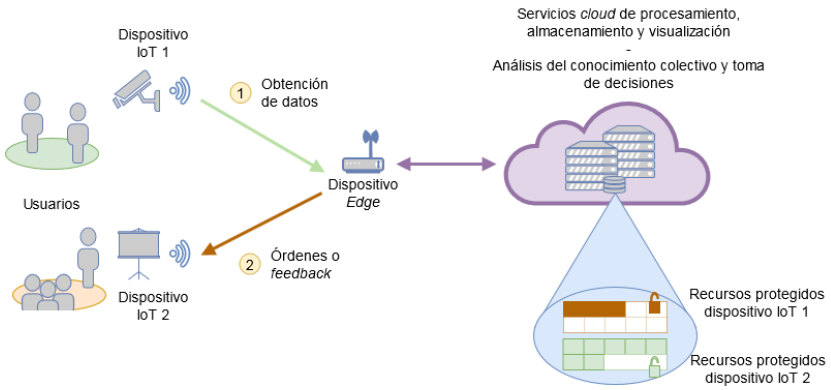

Figura 1.2: Resumen gráfico de la hipótesis de partida de esta tesis

más eficiente de los recursos disponibles en los dispositivos más sencillos. La figura 1.1 muestra este último enfoque que delega parte de la computación en estos dispositivos de borde o *edge* entre ambos mundos y que se conoce como enfoque *edge-centric*.

1.4. Hipótesis de partida

Se ha observado que es necesario plantear un esquema de gestión de identidades que, asumiendo las características y limitaciones intrínsecas de los dispositivos englobados dentro del Internet de las cosas y aprovechándose del enfoque *edge-centric*, permita llevar a cabo una correcta gestión del control de acceso de dichos dispositivos a los servicios *cloud* sin comprometer con ello la seguridad del sistema en su conjunto. Además, debe proporcionar las características necesarias para que los diferentes servicios *cloud*, independientemente del dominio de aplicación, puedan llevar a cabo un direccionamiento y nombrado efectivo para todos y cada uno de los dispositivos registrados en el sistema. Y para que, en los casos en los que sea necesario construir un conocimiento colectivo, pueda hacerse de manera respetuosa con la privacidad de los agentes participantes. Por ello, la hipótesis de partida planteada para la presente tesis doctoral, resumida de manera gráfica en la figura 1.2, es la siguiente:

Es posible proponer mecanismos de seguridad y privacidad para el Internet de las cosas que, teniendo en consideración las restricciones de recursos y capacidades de los dispositivos típicos en este tipo de entornos, sean capaces de aprovechar las oportunidades que los dispositivos de borde (edge), ubicados entre estos dispositivos y los servicios cloud, pueden ofrecer. Más concretamente, es posible definir un esquema de gestión de identidades y accesos que permita llevar a cabo el registro de dichos dispositivos de manera altamente escalable y que sea la base de otros mecanismos necesarios para la interacción segura con servicios cloud (como direccionamiento y nombrado) o para la obtención de conocimiento colectivo de manera respetuosa con la privacidad (como el crowdsensing influenciado).

1.5. Objetivos

El objetivo que esta tesis doctoral pretende conseguir viene determinado por la hipótesis de partida y se resume en:

- Diseñar e implementar un esquema de gestión de identidades y accesos *edge-centric* que tenga en cuenta las limitaciones de recursos y capacidades de los dispositivos englobados dentro del Internet de las cosas y les proporcione un nivel seguridad adecuado para los dominios de aplicación más extendidos para el IoT.

- Proponer un modelo de direccionamiento y nombrado que, apoyándose en el esquema de gestión de identidades y accesos propuesto, permita a los servicios *cloud* disponer de un mecanismo para comunicarse con cualquiera de los dispositivos registrados en dicho esquema, independientemente de los protocolos de comunicación subyacentes.

- Establecer mecanismos que, apoyándose de nuevo en el esquema de gestión de identidades y accesos propuesto, permitan minimizar el impacto negativo que para la privacidad de los usuarios suelen tener las aplicaciones de *crowdsensing*.

Con el fin de materializar estos objetivos principales se plantea el siguiente conjunto de objetivos específicos:

1. Esquema de gestión de identidades y accesos:

    - Definir una arquitectura de referencia de tres capas que sirva como base para toda la investigación.
    - Diseñar un procedimiento de registro o alta en el sistema para los dispositivos de recursos limitados que no suponga incrementar las funcionalidades de seguridad en dichos dispositivos y que permita mantener los niveles de escalabilidad y automatización adecuados para los dominios de aplicación habituales.
    - Diseñar un procedimiento de acceso a recursos protegidos en el *cloud* por parte de los dispositivos que sea compatible con las capacidades limitadas de almacenamiento y procesamiento de dichos dispositivos.
    - Diseñar un procedimiento de acceso en *roaming* por parte de los dispositivos a los recursos protegidos que cubra las características de movilidad que los dispositivos pueden tener en función de los diferentes dominios de aplicación y que pueda servir como método de contingencia.
    - Implementar los procedimientos propuestos de manera que el esquema resultante pueda ser validado y evaluado (desde el punto de vista de su rendimiento, pero también de su seguridad), así como servir de base para el resto de la investigación que se debe realizar.

2. Modelo de direccionamiento y nombrado:

    - Proponer un procedimiento de envío de información u órdenes desde los servicios *cloud* hacia los diferentes dispositivos registrados en el esquema diseñado para la gestión de identidades y accesos que resuelva el direccionamiento mediante un mecanismo jerárquico (compatible con el enfoque *edge-centric seguido en toda la tesis* y basado en eventos para minimizar el consumo de recursos en los dispositivos.

- Definir un esquema de nombrado jerárquico, baso en estándares y escalable, dada la gran cantidad de dispositivos que pueden formar parte de un proyecto o despliegue.
- Validar el modelo propuesto en escenarios complejos donde coexistan múltiples dominios de aplicación, ecosistemas, agentes, etc.

3. Mecanismos de *crowdsensing* respetuosos con la privacidad:

   - Analizar los contextos en los que suelen emplearse aplicaciones de *crowdsensing* para identificar los aspectos en los que esquema de gestión de identidades y accesos diseñado así como el enfoque *edge-centric* empleado pueden ayudar a mejorar los aspectos relacionados con la privacidad.
   - Establecer mecanismos de conocimiento colaborativo que se basen en este análisis y minimicen los riesgos para la privacidad de los participantes.
   - Verificar experimentalmente la validez de los mecanismos propuestos y evaluar los niveles de privacidad que proporcionan.

1.6. Metodología

La metodología propuesta para demostrar la hipótesis de partida planteada en esta tesis y conseguir los objetivos principales descritos es la siguiente:

- Investigación y análisis de los aspectos más importantes de la gestión de identidades y accesos, de los problemas de direccionamiento y nombrado y de las aplicaciones de conocimiento colaborativo en entornos distribuidos, a partir de los trabajos e investigaciones más relevantes.
- Análisis de los aspectos más significativos de los dispositivos englobados dentro del Internet de las cosas así como de las necesidades específicas de sus dominios de aplicación habituales. Tienen especial importancia todos los aspectos relacionados con las limitaciones de recursos disponibles.

- Definición de arquitecturas de referencia compatibles con el enfoque *edge-centric* de esta investigación e identificación de tecnologías, estándares y paradigmas asociados a estas arquitecturas que pueden ser fundamentales para demostrar la hipótesis de partida planteada.

- Propuesta de esquemas, modelos o mecanismos que permitan resolver los problemas estudiados y conseguir los objetivos principales planteados: gestión de identidades y accesos, direccionamiento y nombrado y conocimiento colectivo respetuoso con la privacidad.

- Validación analítica y/o experimental (mediante la implementación de prototipos) de los esquemas, modelos y mecanismos propuestos en casos de uso reales.

- Evaluación de rendimiento y de seguridad/privacidad de las propuestas realizadas.

1.7. Estructura del documento

Esta tesis doctoral está compuesta, además de por el presente capítulo de Introducción, por los siguientes capítulos:

- En el **capítulo 2** se resume el estado del arte, tanto de la gestión de identidades y accesos, como del ámbito del direccionamiento y nombrado. Además, este capítulo incluye el propio estado del arte del denominado Internet de las cosas y cómo los aspectos anteriores, de la gestión de identidades y accesos, y el direccionamiento y nombrado, se encuentran ante diferentes tipos de problemáticas debido a las características de un entorno altamente escalable y heterogéneo en el cual están desplegados dispositivos con recursos limitados tanto de cómputo como de almacenamiento seguro u otros. Finalmente, también se aborda la problemática de la privacidad del usuario que se ve involucrado en entornos en los que los dispositivos funcionan de manera colaborativa para construir un conocimiento conjunto sobre la realidad que les rodea.

- En el **capítulo 3** se expone el diseño propuesto para el esquema de delegación de autorización *edge-centric* que resuelve tanto los problemas de gestión de identidades y accesos, como el de direccionamiento y nombrado en el ecosistema del Internet de las cosas. Para ello, se define la arquitectura de referencia utilizada y las tecnologías y especificaciones fundamentales que dan soporte técnico al esquema planteado. A continuación de presentan los flujos principales que componen el esquema propuesto.

- En el **capítulo 4** se presentan los mecanismos propuestos para las aplicaciones en las que es necesario contar con conocimiento colectivo mediante la colaboración de multitud de dispositivos, también llamado *crowdsensing*, con un enfoque respetuoso con la privacidad del usuario. En este capítulo se aborda el problema del uso de una gran cantidad de dispositivos que obtienen un inmenso volumen de información del mundo físico a través de sus sensores con el fin de, a partir de ella, cambiar el estado de dicho mundo físico o al menos influenciarlo. Por este motivo, se torna tan importante ser respetuoso con la privacidad del usuario, que es quien se encuentra inmerso en dicho mundo físico.

- En el **capítulo 5** se proporcionan detalles sobre la implementación del esquema de delegación de autorización propuesto en el capítulo 3 y sobre su uso para resolver los problemas de direccionamiento, nombrado y conocimiento colectivo respetuoso con la privacidad. Esta implementación permite validar y evaluar todas las soluciones propuestas en los capítulos anteriores mediante su aplicación en dos casos de uso reales, el primero en el dominio de aplicación de la agricultura inteligente y el segundo en el dominio de aplicación de los coches conectados y las carreteras inteligentes. Este capítulo incluye análisis de rendimiento pero también de seguridad y privacidad.

- Con el **capítulo 6** se finaliza este documento, presentando las conclusiones más importantes obtenidas con la realización de esta tesis doctoral y proponiendo líneas de trabajo futuro relacionadas con la investigación realizada.

Capítulo 2

Estado del Arte

A lo largo de este capítulo se expone el estado del arte y fundamentos sobre los que se apoya la investigación realizada durante esta tesis doctoral que versa sobre la gestión de identidades en el Internet de las cosas y la problemática del direccionamiento y del nombrado inherente de dichos dispositivos debido a la alta escalabilidad de su contexto. En primer lugar, dentro del ámbito de la gestión de identidades y accesos, se define los conceptos clave seguidos de los modelos de control de accesos más extendidos para, finalmente, tratar un apartado relacionado con el gobierno y seguridad del dato en sí. A continuación, se aborda el tema del direccionamiento y del nombrado desde un punto de vista general sin entrar en las peculiaridades del Internet de las cosas. Después, se define el contexto en sí de los dispositivos englobados en el Internet de las cosas, tanto sus características intrínsecas, sus limitaciones, capacidades de comunicación y retos que plantean dentro del mundo de las tecnologías de la información. Posteriormente, se trata cómo dichos dispositivos son cubiertos por los dos ámbitos descritos de inicio, es decir, la gestión de identidades y accesos, y el direccionamiento y nombrado. Y finalmente, se desarrolla también la temática relacionada con el uso de dichos dispositivos para beneficiarse del conocimiento colectivo que proporcionan y los problemas de privacidad que se derivan de dicho uso.

Durante el proceso de investigación se ha profundizado en los trabajos y especificaciones más importantes relacionados con la gestión de identidades y accesos en el Internet de las cosas para poder generar un ecosistema robusto y confiable

que, apoyándose en esquemas de federación de identidades, permita cubrir los aspectos más importantes como son la autorización y control de acceso a recursos, así como el direccionamiento y nombrado de los distintos dispositivos en el sistema. Todo ello con el objetivo de detallar y mejorar la comprensión del contexto alrededor del que se asientan los conceptos sobre los que se desarrolla la presente tesis.

2.1. Gestión de identidades y accesos

Dentro del ámbito de la seguridad en tecnologías de la información, existe el campo de la gestión de identidades y accesos, el cual es sumamente importante ya que sobre él pivotan la gran mayoría de campos ya sea para aprovecharse de las capacidades que proporciona o para intentar vulnerar alguna de sus características con vistas a obtener un mayor privilegio o acceso a los recursos en un sistema. En el momento en el que existe más de una entidad interactuando dentro de un sistema ya sea humano, aplicación, servidor u otro, se hace necesario conocer con quién se está interactuando en cada momento dentro del sistema por parte las distintas entidades involucradas en él. La gestión de identidades y accesos [2] es el campo que se encarga de la administración del ciclo de vida de una identidad desde que se aprovisiona en un sistema para poder obtener acceso a los distintos recursos disponibles en él, hasta que finalmente es eliminada y deja de formar parte de dicho sistema.

Dicho de forma sencilla, la gestión de identidades y accesos se encarga de traducir una identidad del mundo físico a una identidad dentro de un sistema informático. Esto permite gobernar el ciclo de vida de dicha identidad en el sistema como se muestra en la figura 2.1 controlando, entre otros, cómo se autentica, a qué está autorizado que acceda la identidad, a qué grupo o grupos pertenece y qué rol o roles desempeña dentro del sistema u organización así como los atributos que definen dicha identidad, cómo se lleva a cabo el rotado de las contraseñas que permite a la identidad autenticarse en el sistema en base a la propia política de contraseñas, o incluso, la aplicación de controles que permitan cubrir el cumplimiento de las políticas de seguridad y buenas prácticas asociadas a la identidad dentro de la organización.

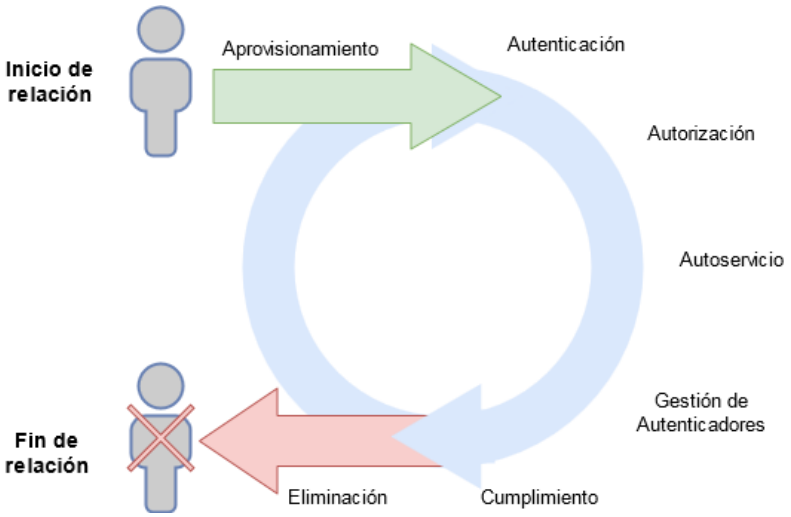

Figura 2.1: Ciclo de vida de una identidad digital

El campo de la gestión de identidades en sí se podría dividir a su vez en los siguientes cinco grandes bloques que cubren distintos aspectos como: el ciclo de vida de las identidades, el control de acceso, la federación de identidades, la gestión de las cuentas privilegiadas y el gobierno del dato.

El bloque del ciclo de vida de las identidades se encargaría principalmente del aprovisionamiento de una identidad en el sistema informático definiendo sus credenciales, atributos y demás información relevante para permitir tanto su alta en el sistema, su gobierno mientras dicha identidad esté vigente, así como su baja del mismo cuando no sea necesaria dicha identidad.

Una vez la identidad está activa dentro del sistema, el siguiente punto clave a tratar sería la gestión y control de acceso, es decir, a qué puede o no acceder una identidad dentro del sistema de tal forma que se defina para ella el mínimo privilegio posible para el correcto desarrollo de sus roles o funciones dentro del sistema.

Siguiendo con la analogía del mundo físico y que en él se busca la unicidad de las identidades, dentro de un sistema informático, una identidad debería ser única también. Es por ello que el bloque de la federación de identidades pone foco en la realización de la sincronización de la identidad a través de los distintos servicios o sistemas sin necesidad de duplicar dicha identidad, es decir, se busca

cubrir cómo se comparte una identidad entre distintos sistemas sin necesidad de que la identidad se duplique en todos y cada uno de ellos y sin embargo, todos los sistemas involucrados confíen en la identidad federada aunque no sean ellos los encargados de custodiarla o de realizar el proceso de autenticación.

Hasta el momento, en los bloques anteriormente descritos existía una correlación uno a uno entre una identidad del mundo físico con su traducción a su identidad dentro de un sistema informático. Sin embargo, en todo sistema suele existir un conjunto de cuentas de administrador o cuentas privilegiadas que se encargan de perfilar el control de acceso a los recursos para todas las demás y que suelen estar compartidas entre distintas entidades del mundo físico. Es por ello que existe dentro del campo de la gestión de identidades y accesos un bloque completo que trata de abordar esta problemática con vistas a ejercer un control sobre dichas cuentas de super-usuarios.

Finalmente, y una vez perfilado al usuario de a qué recurso puede acceder o no, un paso más allá en cuanto al nivel de granularidad del control de acceso sería el propio gobierno del dato. Este gobierno del dato permite describir quién puede tomar qué acciones, sobre qué datos y durante cuanto tiempo, de tal forma que el propio dato en sí pasa a tener identidad propia así como tiempo de vigencia durante el cual es válido y útil para ser consultado, agregado o analizado. Una vez el tiempo de vigencia del dato a expirado, dicho dato no se considera válido para el caso de uso concreto y por lo tanto dejaría de estar accesible.

2.1.1. Conceptos básicos

Todo sistema de gestión de identidades y accesos que se precie se fundamenta siempre de un modo u otro en un conjunto de conceptos básicos que se van extendiendo y adaptando en función de la casuística concreta del sistema pero que siempre tiene la misma base. Lo primero que habría que definir serían las funciones o responsabilidades que pueden tener los distintos actores involucrados dentro del sistema, lo cual se puede categorizar principalmente en dos roles base:

- Usuario: una persona, aplicación, proceso, servidor o cualquier otra entidad que actúe sobre el sistema con vistas a crear, consultar, modificar o eliminar

cualquier configuración, información u otro tipo de activo dentro del mismo, tendría la consideración de incluirse dentro de este rol.

- Recurso: es el activo dentro de un sistema en torno al cual gira la gestión de identidades y accesos con vistas a asegurar que sólo puede ser accedido por las entidades autorizadas para hacerlo, es decir, al tener la consideración de un bien de valor dentro del sistema se intenta salvaguardar de accesos o modificaciones no autorizadas.

Descritos los roles de un sistema de gestión de identidades y accesos, el siguiente punto es tener claro los distintos mecanismos o procesos que el sistema realiza para permitir o denegar el acceso de los distintos usuarios a los recursos. Estos mecanismos serían los siguientes cuatro:

- Identificación: permite establecer un identificador unívoco a la identidad de un usuario con vistas a reconocerla en las subsiguientes interacciones que dicha identidad tenga con el sistema. Este identificador puede ser compartido con la propia identidad mediante una cabecera o *cookie*, o ser el sistema capaz de deducirlo a partir del contexto del usuario basado, por ejemplo, en su IP, su geolocalización, su *User-Agent*, u otras características.

- Autenticación: es el mecanismo por el cual el sistema reta al usuario para comprobar que es quien dice ser. De este modo, el usuario puede ser identificado por el sistema basándose en la premisa de que sólo él debería saber resolver los retos que le plantean para verificar su identidad. Dichos retos se estructuran principalmente en tres categorías:

  - Algo que se sabe: como por ejemplo un usuario y contraseña.
  - Algo que se tiene: como un teléfono móvil donde recibir una clave de un solo uso (OTP - *One Time Password*), DNI electrónico o los datos impresos en una tarjeta de crédito.
  - Algo que se es: este punto está principalmente referido a características biométricas como la huella dactilar, reconocimiento facial o de retina.

- Autorización: una vez se ha autenticado al usuario y se le ha identificado, el proceso que permite definir y controlar a qué recurso puede o no acceder dicho usuario y con qué nivel de privilegio dentro de un sistema sería la denominada autorización o control de acceso.

- Auditoría: consiste en el registro de la trazabilidad de qué usuario ha intentado acceder a qué recurso y en qué momento. Toda esta trazabilidad permite realizar una posterior revisión con vistas a esclarecer cualquier tipo de incidente que haya podido surgir a lo largo del tiempo dentro del sistema y poder llevar a cabo acciones mitigadoras para futuras ocasiones en base a dicha información.

Obviamente, para que pueda existir una relación de confianza entre los distintos actores que interactúan dentro del sistema de gestión de identidades y accesos con independencia de su rol, los procesos descritos anteriormente deben cumplir una serie de características:

- Confidencialidad: es la característica que cubriría el aseguramiento de que la información, tanto en tránsito como en reposo, sólo puede ser leída por aquellas entidades que están autorizadas a ello.

- Integridad: aún cumpliendo la característica anterior, es igual de necesario asegurar que el conjunto de información no ha sido manipulado ni se ha corrompido. Es por ello que tanto la confidencialidad como la integridad siempre suelen ser dos características que van de la mano.

- No repudio: cuando se realiza una acción dentro de un sistema por parte de un usuario, esta característica se encarga de atestiguar que el usuario no pueda negar que fue él quien realizó dicha acción concreta.

- Disponibilidad: esta sería una característica muy importante que permite la existencia de las demás ya que cubriría el caso de que cierta información o recurso concreto se encuentre accesible en el momento justo en el que se requiere acceder a él.

2.1.2. Control de acceso

Como ya se ha comentado al inicio de este capítulo, el control de acceso es uno de los bloques que componen el campo de la gestión de identidades y accesos y se encarga de cubrir quién puede o no acceder a un recurso dentro de un sistema. Este bloque a su vez se podría dividir en dos partes: modelos de control de acceso y su gestión dentro de la organización. Por un lado, los modos de gestión de los distintos modelos de control de acceso [2] dentro de una organización se podrían agrupar en tres tipos: distribuido, centralizado y federado.

El primero de estos tipos, es decir, el distribuido, permite a cada sistema definir sus propias políticas de gestión de acceso para cada usuario. Este modo suele implicar que las diferentes identidades se encuentren replicadas en los distintos sistemas y que la definición de las diferentes políticas de acceso sean propietarias del sistema en el que se definen.

Por otro lado, el modo centralizado se encarga de agrupar las identidades así como el perfilado referido a la gestión de acceso de manera centralizada, lo que fuerza a que los distintos sistemas que proporcionan servicios al usuario deben integrarse con el proveedor de identidades (IdP - *Identity Provider*) para ser capaces de aplicar dichas políticas definidas. Esto permite que el gobierno de las políticas de control de acceso se lleve a cabo de manera centralizada y se apliquen en todos los sistemas de la organización por igual aunque como punto negativo de este enfoque, este proveedor de identidades centralizado pasa a ser el único punto de fallo del sistema de gestión de identidades.

Finalmente, el modo federado consiste en dar un paso más allá sobre el sistema centralizado, es decir, utilizando protocolos de federación de identidades focalizados en el control de acceso como OAuth 2.0 [3], la capa de autorización y control de acceso que se apoya en la interpretación de las políticas definidas para ello queda completamente desacoplada de los diferentes sistemas, los cuales delegan dicha gestión del control de acceso en los servidores de autorización de la organización o incluso, en los servidores de autorización existentes entre distintas organizaciones entre las cuales se ha establecido una relación de confianza entre ellas. Este último punto permite suplir el punto negativo identificado en los sistemas centralizados relacionado con ser éstos el único punto de fallo del sistema.

Una vez categorizados los distintos modos de gestión de los modelos de control de acceso, el siguiente paso es tratar los modelos de control de acceso en sí. A continuación se detallan los cinco modelos de control de acceso más significativos:

- Control de acceso discrecional (DAC [4] - *Discretionary Access Control*): es un método de control de acceso que se basa en las propias restricciones o políticas de acceso que el dueño del recurso establezca para indicar quién puede acceder o no sobre sus propios recursos y de qué modo, es decir, cada recurso protegido puede tener distintas restricciones en base a las necesidades que especifique el dueño de dicho recurso.
  Un caso concreto de aplicación de este modelo de control de acceso estaría implementado en los sistemas Unix, donde los recursos del sistema como ficheros, directorios u otros, cuentan con ternas de atributos de lectura (r), escritura (w), y ejecución (x). Estos permisos son asignados por el propietario del recurso (*user*) tanto para él, como para el grupo al que pertenece (*group*) y para terceros (*others*) los cuales son usuarios no propietarios y no pertenecientes a su grupo. Otro ejemplo significativo de este modelo de control de acceso sería el de los recursos existentes en Google Drive, donde el propietario del recurso puede permitir a otros usuarios la lectura de documentos, la realización de comentarios sobre ellos, o incluso, la edición completa de los mismos.

- Control de acceso obligatorio (MAC [5] - *Mandatory Access Control*): es un modelo de control de acceso que complementa a todos los demás ya que añade una capa más de seguridad basada en el etiquetado de los recursos existentes en el sistema. Sólo los usuarios privilegiados del sistema tienen la responsabilidad de fijar las políticas centrales a aplicar según el etiquetado establecido sobre los distintos recursos. Dicho etiquetado se basa en dos criterios: la clasificación de la información y la categoría a la que pertenece. En primera instancia, la clasificación de la información está basada en los distintos niveles de sensibilidad que puede tener dicha información dentro de la organización, es decir, información pública, privada, confidencial, secreta u otros. Por otro lado, la categoría a la que pertenece la información

está más focalizada a la propia unidad organizativa dentro de la organización a la que pertenece como puede ser el departamento del que depende o incluso, el proyecto que la ha generado. De este modo, cualquier operación de un usuario sobre un recurso será previa comprobación de las diferentes etiquetas y aplicando las políticas establecidas para determinar si la operación está permitida o no. En contraste con el mecanismo de control de acceso discrecional, el usuario puede modificar permisos y etiquetas pero no puede fijar controles de acceso que supongan una violación de las políticas centralizadas del sistema.

- Control de acceso basado en roles (RBAC [6] - *Role-Based Access Control*): es un modelo de control de acceso que aporta mayor versatilidad si lo comparamos con el modelo de control de acceso discrecional, ya que éste último no proporciona la granularidad suficiente para permitir una segmentación definida y estructurada en un sistema complejo con multitud de usuarios y funciones. Por tanto, el control de acceso basado en roles consiste en la definición de perfiles (o también llamados roles) a los que se les atribuyen una serie de características que aplican sobre los permisos y acciones que pueden llevar a cabo dentro de la organización, incluyendo incluso, el control sobre otros perfiles. Este modelo está muy extendido en el ámbito de la industria en organizaciones con un gran número de usuarios donde se integran distintos grupos de trabajo o departamentos con funciones diferenciadas, como por ejemplo; departamento de sistemas, departamento de desarrollo, comercial, servicios jurídicos, etc. Es por esto que con este mecanismo se segmenta y se organiza de forma eficaz el acceso a los recursos en base a las funciones y tareas acometidas por cada uno de ellos dentro de la organización.

- Control de acceso basado en atributos (ABAC [7] - *Attribute-Based Access Control*): permite otorgar el acceso a recursos a los distintos usuarios basándose para ello en políticas que combinan diferentes atributos. Dichas políticas pueden usar cualquier tipo de atributos del usuario, de los propios recursos a los que se intenta acceder, del entorno o contexto alrededor del acceso por parte del usuario como su IP, su geolocalización u otras ca-

racterísticas, etc. Este modelo se apoya en reglas condicionales de carácter lógico de cierto o falso que pueden llegar a ser altamente complejas llegando a evaluar multitud de diferentes atributos para conceder o no un acceso. Sin embargo, este dinamismo a la hora de evaluar la concesión de acceso permite realizar una evaluación sensible al contexto e inteligente frente a los riesgos, permitiendo definir políticas de control de acceso que incluyen atributos específicos de muchos sistemas de información diferentes con vistas a resolver una autorización y lograr un cumplimiento normativo eficiente.
Para llevar a cabo la materialización de este modelo de control acceso, se utiliza el estándar XACML [8] (*eXtensible Access Control Markup Language*). Dicho estándar XACML consiste en un lenguaje declarativo de políticas de control de acceso de grano fino, una arquitectura, y un modelo de procesamiento de cómo evaluar el acceso de las peticiones en base a las reglas definidas. El modelo XACML soporta y fomenta la separación entre la decisión de acceso y el propio recurso a consumir, ya que cuando las decisiones de control de acceso se incluyen dentro de las aplicaciones, es muy difícil de actualizar los diferentes criterios de decisión si el gobierno de las políticas cambia. Sin embargo, cuando una aplicación se desacopla de las decisiones de acceso, las políticas pueden ser actualizadas al vuelo sin necesidad de afectación al servicio. En la figura 2.2 se muestra la arquitectura de referencia, la cual se basa en los siguientes componentes:

- PEP (*Policy Enforcement Point*): es el punto de forzado de las políticas ya que se encarga de interceptar la petición de acceso del usuario a un recurso, hacer una petición al PDP para obtener la decisión de permitir o denegar el acceso y actúa en base a la decisión recibida.
- PDP (*Policy Decision Point*): es el punto de decisión de las políticas y se encarga de evaluar las peticiones de acceso contra las políticas de control de acceso definidas antes de emitir decisiones de acceso.
- PIP (*Policy Information Point*): es el punto de información de la política el cual actúa como fuente de valores de atributos para enriquecer los datos para que el PDP pueda tomar la decisión correcta.
- PAP (*Policy Administration Point*): es el punto de administración de

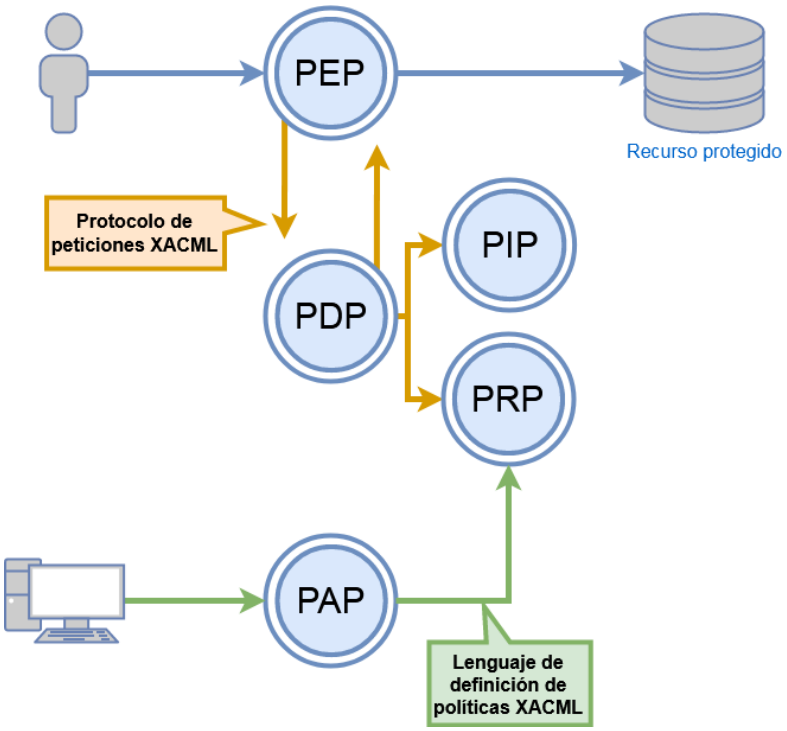

Figura 2.2: Arquitectura XACML de referencia en modelos ABAC

las políticas de acceso, es decir, donde se crean, se versionan o eliminan.

- PRP (*Policy Retrieval Point*): es el punto de recuperación de políticas XACML, es decir, donde se almacenan las políticas para su administración desde el PAP.

- Control acceso centrado en roles y basado en atributos (RABAC [9] - *Role-Centric Attribute-Based Access Control*): es un modelo de control de acceso que aunque a día de hoy no está siendo utilizado por la industria, está surgiendo con el fin de mitigar los dos problemas base de los dos últimos modelos de control de acceso mencionados: la problemática de la gestión del gran número roles que se pueden generar dentro de una organización basada en el uso del modelo de control de acceso basado en roles, y el gobierno de la ingente cantidad de reglas XACML en un modelo complejo de control de acceso basado en atributos. Por este motivo, este modelo añade al

modelo basado en roles una serie de atributos tanto sobre los usuarios como sobre los recursos, y una política de filtrado de permisos (PFP - *Permission Filtering Policy*) que se encarga de restringir el conjunto de permisos disponibles para un usuario dentro de una sesión en base a sus propios atributos de usuario y a los del recurso dentro de dicha sesión. Dicho de otro modo, los permisos de un usuario ya no dependen sólo del rol (o roles) que ostenta sino que además, los propios atributos recopilados en la sesión en curso en la cual requiere acceso de los diferentes recursos son los que definen el máximo conjunto de privilegios posibles para ese usuario en dicha sesión.

La evolución de los modelos de control de acceso deja entrever cómo las características estáticas van siendo relegadas a un segundo plano y emerge cada vez más la gestión centrada en los propios atributos del contexto de los dispositivos en el acceso. Esto provoca que a día de hoy, surjan como una solución natural para maximizar el uso de los diferentes sensores y actuadores disponibles los modelos de control de acceso basados en el contexto (*Context Aware Access Control*), los cuales permiten, además de realizar un control de acceso basado en atributos, ser una fuente importante para extraer información de contexto del mundo físico, interpretarla, y usarla para adaptar su funcionalidad al contexto actual de uso [10].

Por tanto, al ser necesario cada vez más información de contexto tanto para la parte funcional de los diferentes aplicativos como de los modelos de control de acceso, éstos comienzan a su vez a apoyarse en modelos de control de acceso basados en *token* (TBAC - *Token-Based Access Control*) ya que dichos modelos proporcionan una gestión distribuida y confiable del acceso de una manera eficiente, flexible y tolerante a fallos. Por este motivo, multitud de trabajos dentro de los modelos de control de acceso basados en contexto y/o en atributos que usan esta extensión con *tokens* se enfocan en reforzar dichos *tokens* apoyándose en el uso de la criptografía con el fin de incrementar el nivel de confiabilidad en dicho enfoque [11–13].

2.1.3. Gobierno del dato

Antes de comenzar a tratar qué es el gobierno del dato en sí, se torna necesario clarificar dos términos que muchas veces se confunden dentro de este ámbito: la

gestión del dato (en inglés, *Data Management*) y el gobierno del dato (en inglés, *Data Governance*). Por un lado, según la asociación DAMA (*Data Management Association*) [14], la gestión del dato cubriría el despliegue, ejecución y supervisión de planes y políticas con vistas a controlar, proteger y enriquecer el valor del dato y de los activos de información dentro de una organización. Dicho de otro modo, la gestión del dato se focaliza en la definición de los propios datos, en cómo se estructuran y almacenan, y cómo se mueven entre los diferentes sistemas. Por otro lado, según el instituto DGI (*Data Governance Institute*) [15], el gobierno del dato es el marco para definir el conjunto de derechos y responsabilidades sobre el uso deseable de los datos, promoviendo dicho comportamiento deseable, y desarrollando e implementando el conjunto de políticas y estándares acordes con la misión, estrategia, valores, normas y cultura de la organización a la que pertenecen. En definitiva y de manera resumida, el gobierno del dato se focaliza en el ejercicio de la autoridad y control sobre la gestión de los activos de información con vistas a cubrir la disponibilidad, usabilidad, integridad y seguridad de los datos de una organización.

Para poder implementar un correcto gobierno del dato, cada organización puede definir un conjunto diferente de políticas alrededor de este activo tan importante, pero en mayor o menor medida, todas estas políticas intentan cubrir siempre los siguientes aspectos [16]: la propia identificación y clasificación de los datos, la gestión del ciclo de vida del dato, el incremento de las medidas de autorización y control de acceso, y un mayor foco en la concienciación y el cumplimiento normativo.

En primer lugar, una correcta clasificación y organización de dichos activos permite llevar a cabo una precisa aplicación de la seguridad sobre aquellos datos considerados sensibles de tal forma que se reduzcan los riesgos de seguridad y privacidad asociados con independencia de la ubicación geográfica donde se almacenen, ya sea en el centro de procesamiento de datos de la propia empresa o en el *cloud*.

Por otro lado, de la misma manera que una identidad, desde el nacimiento del dato éste se debe clasificar y almacenar de acuerdo a sus propiedades particulares dentro de la organización para que la protección que se establezca sobre él le acompañe a lo largo de su tiempo de vida útil. Esto obviamente incluye el enten-

dimiento del tipo del dato, sobre qué flujos será relevante su uso y cuáles serían los vectores de fuga de dicho dato, ubicación en la que se persiste y estado (cifrado, tokenizado, ofuscado, o en claro, etc.), y todo ello siempre focalizado en un enfoque basado en riesgos, es decir, no todos los datos son creados de la misma manera y por eso, el programa de gobierno del dato dentro de una organización debería estar focalizado en la protección de la información sensible en lugar de en toda la información disponible en dicha organización.

Todo esto provoca que sea necesario la revisión y refuerzo de dichos controles de acceso para garantizar la seguridad de los datos. Esto es así ya que los empleados dentro de una organización sólo deberían tener acceso a los datos necesarios para cumplir con sus responsabilidades o roles. Sin embargo, cuando el programa de gobierno del dato se lleva a cabo a posteriori de la implantación del sistema de gestión de identidades en una organización, surge el reto de intentar limitar el conjunto de roles de accesos de los usuarios a los datos ya que dichas restricciones son definidas por defecto de una forma muy amplia. Por este motivo, siendo el dato un activo crítico para cualquier organización, se debe no sólo aplicar políticas de control de acceso a dicho dato, sino también a la propia red con el fin de evitar que dispositivos no autorizados intentan obtener información sensible.

Sin embargo, en muchas ocasiones los programas de concienciación de las organizaciones no son correctamente puestos a prueba para validar el conocimiento y cumplimiento de las políticas de la organización por parte de los empleados. Por tanto, se suelen llevar a cabo campañas de *phishing* e ingeniería social para determinar el grado de concienciación de los empleados y de este modo, poder reforzar y asegurar aquellos riesgos latentes que podrían provocar la pérdida de información por cualquiera de los diferentes vectores posibles.

El crecimiento exponencial del uso de los paradigmas *cloud* junto a la gran cantidad de datos que se recopilan del mundo físico a través de los diferentes sensores provocan que el dato tome aún más importancia dentro de las diferentes organizaciones a la hora de tomar decisiones estratégicas. Por este motivo, a día de hoy se está poniendo principalmente foco en generar marcos de control que evalúan los niveles de madurez de una organización en el gobierno del dato sobre entornos *cloud* [17, 18], ya que los modelos tradicionales no encajan en dichos entornos de características y tecnologías cambiantes que sin una gestión o evalua-

ción adecuada puede desembocar en una brecha de seguridad que implique fuga de información lo que supondría un desastre para el negocio.

2.1.4. Direccionamiento y nombrado

El direccionamiento y el nombrado son dos términos dentro de un sistema de tecnologías de la información muy ligados mutuamente que persiguen un objetivo similar: la interconexión de las diferentes partes que conforman dicho sistema. Ambos términos están muy ligados a su vez con la propia identificación derivada de la gestión de identidades, ya que la interconexión entre las partes en sí es un medio para conseguir el fin último del dominio de aplicación concreto del sistema.

El direccionamiento es el método por el cual se puede establecer quién es el emisor y quién es el receptor de un mensaje, y realizar la transmisión de los datos entre ambos dentro de una red de comunicaciones. Dentro del alcance de este concepto se abstrae la propia definición de cómo se representa la identidad de cada uno de los dispositivos involucrados en la comunicación, que puede ser mediante representación lógica o numérica, y simplemente se pone foco en el uso de esas identidades por ambas partes para llevar a cabo la transmisión del mensaje. El direccionamiento se apoya a su vez sobre el concepto de la conectividad, el cual es la capacidad o disponibilidad que tiene un dispositivo para establecer una conexión ya sea con otro dispositivo o con una red sin entrar en identificación de dispositivos o transmisión de mensajes.

Por otro lado, el nombrado es el conjunto de técnicas que se utilizan para realizar la representación de una forma lógica y entendible por los humanos de los dispositivos que conforman un sistema, entendiendo por dispositivo su amplia acepción (dispositivos del Internet de las cosas, móviles, ordenadores portátiles, servidores, etc.), con el fin de identificarlos de manera unívoca. Dicho de otro modo, el nombrado de dispositivos intenta abstraerse de las peculiaridades de cada dispositivo, incluyendo las particularidades de los protocolos de red como direcciones IP, para realizar una representación del mismo que encaje dentro el dominio de aplicación en el que está desplegado. Obviamente, los esquemas de nombrado actuales son una evolución natural de las arquitecturas tradicionales IP, gracias a los cuales se pueden nombrar objetos en general como ficheros, datos u otros,

en vez de puntos concretos y accesibles dentro de la red de comunicación con direcciones IP. Esta característica ha desembocado en la creación de soluciones o redes centradas en nombres o también llamadas en inglés *Name-Oriented Networks* [19–21]. En esta clase de redes, las interacciones están dirigidas por los propios consumidores ya que éstos sólo necesitan conocer el nombre concreto del recurso dentro del sistema al que quieren acceder para hacerlo. Esto propicia que dichos recursos, al disponer de esa identificación unívoca, puedan ser protegidos individualmente siguiendo un enfoque centrado en el dato. Finalmente, dicha gestión de grano fino centrada en el dato ha propiciado la creación de mecanismos ligeros de nombrado que soportan las limitaciones de recursos inherentes dentro de los dispositivos del Internet de las cosas sin conllevar la pérdida de funcionalidad dentro del sistema [22, 23].

2.2. Internet de la cosas

2.2.1. Dispositivos

Los dispositivos englobados dentro del denominado Internet de las cosas (*Internet of Things* - IoT), se podrían describir a muy alto nivel, como ya indicó la ITU (*International Telecommunication Union*) en 2005 [24] en su visión sobre este término acuñado por ellos mismos, como: ”una nueva dimensión ha sido añadida al mundo de las tecnologías de la información y las comunicaciones: a cualquier hora, en cualquier lugar con conexión por cualquiera, podremos conectarnos con cualquier cosa. Las conexiones se multiplicarán y se creará una nueva y dinámica red de redes; un Internet de las cosas”. Cabe destacar la importancia dentro de esta visión del todo, es decir, de no marcar ningún tipo de limitación a qué puede estar o no conectado dentro de este mundo del Internet de las cosas, lo que añade esa nueva dimensión IoT a las arquitecturas tradicionales como muestra la figura 2.3. Es por ello, que los dominios de aplicación del Internet de las cosas es enorme, siendo los tres mayores dominios de aplicación el industrial, incluyendo no sólo los procesos industriales sino también la agricultura y la ganadería, el médico-sanitario, y el de las ciudades inteligentes con casos de uso, entre otros, de movilidad y turismo.

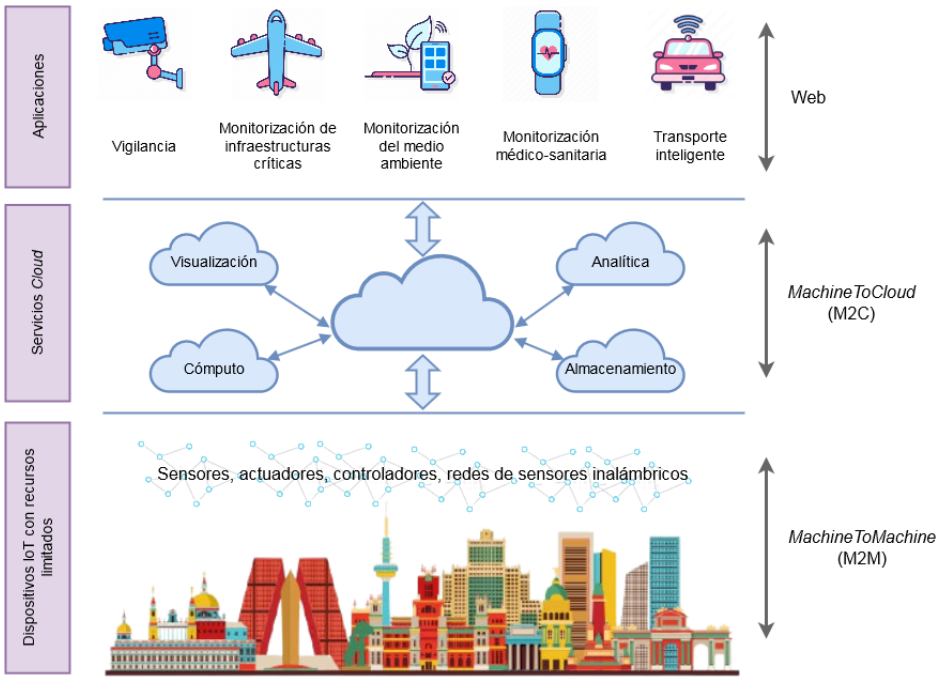

Figura 2.3: Enfoque tradicional de arquitecturas IoT

Esta enorme aplicabilidad ha provocado que las instituciones de estandarización más importantes como son NIST [25], ISO [26], AIOTI [27], ITU [28] e IEEE [29], hayan definido distintos estándares en los que se especifica la nomenclatura, el rol o roles que cubren dichos dispositivos dentro de las arquitecturas de despliegue IoT para dar soporte a los diferentes dominios de aplicación en función de las funcionalidades que proporcionan al sistema o las capacidades de cómputo y comunicación que disponen, etc. Aunque cada estándar define los dispositivos IoT con diferente nivel de detalle y cualidades, se identifica como punto en común entre todos ellos la existencia de dos grandes bloques funcionales en los que se pueden dividir principalmente los dispositivos IoT:

- Sensores: son los dispositivos encargados de recolectar información de su entorno, como por ejemplo, temperatura, humedad, sonido, movimiento, etc. Estos dispositivos suelen ser los más limitados de todos e intentan llevar a cabo el uso más eficiente posible de batería para alargar su tiempo de vida. Por este motivo, toda la información que recopilan suelen enviarla a

sistemas con más capacidad de cómputo para su procesamiento como el *cloud*.

- Actuadores: su función principal es la de interactuar con el mundo físico que les rodea para cambiar su estado o influenciar en él. La necesidad de provocar dicho cambio en el mundo físico suele derivarse de la recopilación de información llevada a cabo por los diferentes sensores y las decisiones tomadas tras el procesamiento de la información recabada por parte del *cloud*. Los ámbitos de aplicación de estos actuadores están muy ligados a los sensores que han detectado la necesidad de que dichos actuadores tomen parte del entorno, es decir, y siguiendo con el ejemplo indicado en el punto de los sensores, los actuadores intentarían adecuar temperatura, humedad, alertar de sonido o movimiento no permitidos, etc.

Aún existiendo dichos bloques diferenciados, los dispositivos de uno y otro bloque comparten características entre ellos. Esto es así ya que los propios dispositivos en sí pueden pertenecer a un dominio concreto o cubrir múltiples dominios de aplicación, por lo que dependiendo del caso, éstos pueden mostrar características específicas del dominio al que pertenecen o en caso de cubrir múltiples, mostrar unas características más genéricas. Desde un punto de vista general de las diferentes características de los distintos dispositivos que forman parte de uno u otro dominio, podríamos destacar [30]:

- Heterogeneidad: al no estar definido el conjunto específico de dispositivos englobados dentro del Internet de las cosas, existe una gran variedad de dispositivos de distintos fabricantes, o desarrollados por particulares gracias a la explosión del hardware libre, que propicia la coexistencia dentro de los sistemas de multitud de tecnologías, servicios y ecosistemas diferentes. Estas características y peculiaridades deben ser abordadas a nivel arquitectural dentro de los distintos dominios de aplicación incluyendo a nivel de nombrado, identificación, direccionamiento y comunicaciones, entre otros.

- Escalabilidad: esta característica va muy ligada a la de la heterogeneidad. La gran variedad de dispositivos que pueden incorporarse dentro de un sistema con vistas a cubrir el objetivo del dominio en el que se despliegan

propicia que el crecimiento del número de dispositivos en él pueda llegar a ser incluso exponencial. Se podría llegar a encontrar similitudes entre la escalabilidad de los dispositivos de bajo coste en los sistemas del Internet de las cosas con la escalabilidad horizontal existente a día de hoy en los entornos *cloud* gracias al alto grado de automatización y despliegue de entornos virtualizados.

- Ahorro de costes: es el eje principal del Internet de las cosas ya que busca optimizar el coste relacionado con el desarrollo, instalación y mantenimiento de los distintos dispositivos dentro de los diferentes dominios de aplicación. En muchas ocasiones, los dispositivos son desarrollados desde cero para cubrir las necesidades específicas del dominio concreto para el que se crean incluyendo sólo las características mínimas que necesitan dentro de dicho dominio. Esto les permite buscar la eficiencia desde el punto de vista de la energía consumida por dicho dispositivo, una característica muy importante dadas las especificaciones de muchos de los casos de uso en los que se despliegan dichos dispositivos.

- Capacidades de autogestión: debido a que en la mayor parte de los casos de uso de aplicación de los dispositivos del Internet de las cosas se requiere de una baja o inexistente interacción humana, los dispositivos tienen que ser de facto capaces de disponer de cierto grado de autonomía, organizándose y adaptándose por sí mismo a los diferentes escenarios que se les plantean, y en ocasiones incluso, responder a los diferentes estímulos que reciben al estar embebidos en el mundo físico en función del conjunto de información que recaban de su entorno.

- Calidad del servicio (QoS - *Quality of Service*): obviamente, en el momento en el que dichos dispositivos toman parte del entorno físico en el que están desplegados o su información sirve para ser agregada con vistas a tomar decisiones sobre él, se convierte en un requisito de obligado cumplimiento por parte de estos dispositivos la calidad del servicio proporcionado por ellos al estar incluidos dentro de un ecosistema, en muchas veces sensible, que gestiona datos en tiempo real.

- Configuración segura: por último, pero no por ello menos importante, es la relevancia de que los dispositivos cubran características de seguridad desde el punto de vista de las comunicaciones, autenticación, autorización y control de acceso, integridad de los datos y de ellos mismos como dispositivos, privacidad de los usuarios y de su información sensible que gestionan, y proporcionen además, la confiabilidad suficiente dentro del entorno en el que están desplegados incluyendo en sus interrelaciones con el resto de partes involucradas en dicho entorno. Aunque todos estos aspectos son sumamente importantes, la protección de los dispositivos, servicios y aplicaciones del Internet de las cosas aún plantean una gran cantidad de retos que resolver desde el punto de vista, entre otros, del control de acceso no autorizado a la información dentro de los sistemas [31]. En este aspecto, las características de heterogeneidad y escalabilidad ya mencionadas chocan de frente con el modelo tradicional de confianza para campos como el de la identificación, autenticación y autorización de los dispositivos. Por este motivo, el nuevo modelo de seguridad que se aplica en estos casos intenta buscar un enfoque que se aproveche de las características ya mencionadas teniendo en consideración las especificaciones de los diferentes escenarios [32] proponiendo soluciones escalables, ligeras, eficientes y robustas.

Todas estas características de los dispositivos y el entramado de interconexión entre ellos materializado con vistas a cubrir un caso de uso concreto conforman un conjunto de redes dinámicas en la que los diferentes nodos pueden conectarse y desconectarse en función de su cobertura, estado de batería u otros factores. Dado este dinamismo y complejidad en la propia red, surgen grandes retos vinculados con el direccionamiento y el nombrado de los propios dispositivos, los cuales son esenciales cubrir ya que proporcionan la capacidad de acceso por parte de dichos dispositivos a los distintos servicios y recursos desplegados en el *cloud*. Éstos deben ser abordados en muchas ocasiones por algún tipo de *middleware* [33] que les apantalle y reduzca la complejidad de la conexión directa entre ambos mundos para el intercambio de información.

Con el fin de estandarizar una arquitectura de referencia que cubra esas características que proporciona un *middleware* que apantalla e independiza el mundo

del Internet de las cosas y del *cloud* (figura 1.1), surge el paradigma de la computación de borde (*Edge computing*) o computación en la niebla (*Fog computing*). Este paradigma introduce dentro del ecosistema un conjunto de dispositivos como controladores, *hubs* o *gateways* que se ubican lo más cercanos al conjunto de dispositivos del Internet de las cosas con vistas a apantallar por un lado, al *cloud* de la heterogeneidad de tecnologías y protocolos de comunicaciones subyacentes, y por otro, para habilita a los dispositivos la capacidad de comunicación e interacción con el *cloud* [34, 35]. Dicho de otro modo, estos nuevos dispositivos de borde son usados como intermediarios lógicos o *proxies* entre ambos mundos lo que permite delegar en ellos características de seguridad e interoperabilidad que no pueden ser abordadas o cubiertas por los dispositivos de capacidades restringidas y que por ende, incrementa los niveles de seguridad dentro del sistema [36].

2.2.2. Comunicaciones

La propiedad principal que define los dispositivos englobados dentro del Internet de las cosas es claramente la de la creación de dispositivos de bajo coste adaptados al dominio de aplicación en el que se van a desplegar. Obviamente esta peculiaridad mediante la cual se desechan funcionalidades o características no utilizables por el dispositivo dentro del dominio en cuestión están claramente enfocadas en la optimización del consumo de batería por parte de dicho dispositivo. Entre dichas funcionalidades que en muchas ocasiones se pueden ver afectadas podrían englobarse, entre otras, almacenamiento seguro, capacidad de cómputo, limitaciones de tamaño y peso que fuerzan a tener una batería más reducida o incluso, capacidades de conectividad.

Este último aspecto relacionado con las capacidades de conectividad de un dispositivo restringido en comparación con otros dispositivos más robustos como móviles, ordenadores portátiles o servidores, fuerza a que los protocolos de comunicación utilizados dentro del ecosistema del Internet de las cosas cumplan una serie de características para que puedan ser útiles bajo dichas restricciones. Entre las características que suelen disponibilizar dichos protocolos de comunicaciones [30] destacan la gestión eficiente del tráfico de red y el soporte de redes dinámicas.

Por un lado, la gestión eficiente del tráfico de red está principalmente centrada en los diferentes requisitos de tiempos y transmisión de datos soportado por los propios dispositivos para cubrir su función dentro del sistema. Estos protocolos buscan gestionar el tiempo de una manera eficiente focalizándose en soportar la tolerancia a los retardos en la comunicación y en incorporar características de baja latencia. Además, para la transmisión de datos se suelen habilitar distintos perfiles que buscan establecer una baja cantidad de datos en la transmisión, gestionar periodos entre dos transmisiones o incluso, aplicar un enfoque de transmisiones continuas.

Por otro lado, dadas las características de un ecosistema de dispositivos y cómo puede cambiar su esquema de red a lo largo del tiempo, los protocolos de comunicaciones se focalizan en ser capaces de cubrir la gestión dinámica de las propias redes soportando características como la selección de rutas de transmisión, la transmisión masiva entre dispositivos cercanos, los modos de direccionamiento que permitan reducir la carga de red, y la garantía de transmisión de mensajes aun cuando los propios dispositivos se mueven dentro del sistema debido a sus características dentro del dominio de aplicación concreto o se encuentran en *standby* por ahorro de energía.

Estas características de dinamismo y optimización del uso del tráfico de red obviamente no están soportadas por protocolos de aplicación orientados a sesión como por ejemplo HTTP [37] (*Hypertext Transfer Protocol*). Dicho protocolo es usado por dispositivos más robustos como móviles, ordenadores portátiles o servidores, ya que posibilita el uso de características y protocolos de seguridad que permiten robustecer los sistemas y las comunicaciones. Por este motivo, el uso de protocolos de comunicación ligeros en el Internet de las cosas es algo necesario. A día de hoy, los dos protocolos de comunicación ligeros más extendidos son:

- CoAP [38] (*Constrained Application Protocol*): es un protocolo similar a HTTP que sigue el modelo cliente/servidor y que permite la transferencia de información entre dispositivos con características limitadas y sobre redes de baja potencia o con pérdida de conectividad. Este protocolo está principalmente diseñado para la interacción máquina a máquina (M2M - *Machine-to-Machine*) por lo cual, los roles de cliente y servidor entre ambas

máquinas, dependiendo del caso, son intercambiados entre los dispositivos. Las peticiones de CoAP son equivalentes a las de HTTP. En este sentido, el cliente envía una solicitud con una acción o método sobre un recurso identificado mediante URI [39] al servidor, el cual responde al cliente con un código de respuesta pudiendo incluir además en dicha respuesta, la representación del recurso como sucede en HTTP.
Aunque CoAP proporciona interacción solicitud-respuesta entre aplicaciones, soporta el descubrimiento de servicios y recursos, e incluye los conceptos clave del mundo web como URI e *Internet Media Types* [40] como HTTP, a diferencia de HTTP, CoAP intercambia sus mensajes de forma asíncrona y sobre UDP [41]. Esto le permite reducir la sobrecarga de la red, soportar envíos *multicast* y simplificar las comunicaciones gracias a la definición de cuatro tipo de mensajes que permiten incluso, de manera opcional, confiabilidad en la entrega sobre UDP con gestión de reintentos.

- MQTT [42] (*Message Queuing Telemetry Transport*): es un protocolo de mensajería cliente/servidor que se basa en el patrón de publicación/suscripción y que se ejecuta sobre TCP [43] o sobre cualquier otro protocolo de red que soporte orden en la entrega de paquetes sin pérdida, y proporcione la capacidad de comunicación bidireccional entre el cliente y el servidor. Estas características permiten la distribución de mensajes en modo *multicast*, baja sobrecarga en el uso de la red, y mecanismos para detectar desconexiones anómalas por parte de los dispositivos incluidos dentro del sistema. Esto es así ya que permite tres tipos diferente de confiabilidad en la entrega de mensajes:

  - Como máximo una vez: este enfoque está pensado principalmente para casos en los que puede ocurrir pérdida de mensajes y se intenta hacer el mejor esfuerzo para el envío y recepción de los mismos.
  - Al menos una vez: donde se asegura la recepción del mensaje por parte del o de los destinatarios pero que puede darse el caso de mensajes duplicados.
  - Exactamente una vez: donde los mensajes, dado el caso de uso, deben

llegar exactamente una vez sin mensajes duplicados.

2.2.3. Paradigmas de computación de referencia

Aunque a lo largo del presente capítulo ya se ha mencionado alguna arquitectura de referencia englobada dentro del marco del Internet de las cosas, a día de hoy existen multitud de ellas derivadas de diferentes paradigmas de computación distribuidos que intentan cubrir las necesidades de los múltiples contextos y dominios de aplicación donde se despliega la gran heterogeneidad de dispositivos de IoT. A lo largo de esta sección se describen los paradigmas de computación de referencia más destacados dentro del ámbito del Internet de las cosas:

- Paradigma de computación *Fog* (o computación en la niebla) [44, 45]: la idea principal de la computación *Fog* consiste en extender los servicios y funcionalidades que proporciona el *cloud* a una infraestructura mucho más cercana de los usuarios o dispositivos finales. Entre dichas funcionalidades se pueden incluir el almacenamiento, la capacidad de cómputo, bases de datos, capacidades de integración y seguridad, y la administración de los dispositivos finales aprovechándose de su proximidad a ese extremo de la red. Esto permite abordar los cuellos de botella de la conectividad disminiyendo la latencia de red, mejorar la seguridad y la privacidad de los dispositivos con recursos limitados en sus conexiones con el *cloud*, mejorar la escalabilidad y la interoperabilidad dentro del sistema, y posibilitar una mejor respuesta en tiempo real que los modelos basados en conexiones directas con el *cloud*. Este tipo de paradigma de computación distribuida se presenta por tanto como una arquitectura de tres capas [46, 47] en las cuales, en un extremo tenemos la capa de servicios *cloud*, en el otro la capa de servicios IoT que son los que se encuentran embebidos en el mundo físico, y finalmente, la capa intermedia introducida por este paradigma que sería la de los servicios *Fog*.

- Paradigma de computación *Edge* (o computación en el borde) [44, 45]: la computación *Edge* descansa en los dispositivos de borde o también llamados dispositivos *edge*, los cuales realizan tareas de computación con el fin

Tabla 2.1: Comparación entre computación *Fog* y computación *Edge*

Características	**Computación *Edge***	**Computación *Fog***
Ubicación del dato	Dispositivos *edge*	Dispositivos *edge* de red
Múltiples aplicaciones IoT	No soportado	Soportado
Foco	Dispositivos finales IoT	Infraestructura
Proximidad con los dispositivos finales	Entre ellos	Cerca de ellos
Capacidad de recursos	Muy limitados	Limitados

de reducir la latencia de red y el ancho de banda entre los dispositivos embebidos en el mundo físico y los servicios del *cloud*. Esto mejora obviamente la capacidad de respuesta de este tipo de arquitecturas acercándose lo máximo posible al procesamiento en tiempo real. Una particularidad de estos dispositivos de borde es que no sólo actúan como consumidores o traductores de datos el *cloud* para los dispositivos finales, sino que también pueden ser ellos los productores de datos, por lo que dichos dispositivos no sólo solicitan servicios al *cloud* sino que también le pueden brindar sus propias capacidades. Este enfoque de computación en muchas ocasiones se confunde con la computación *Fog* ya que ambos persiguen el mismo objetivo principal de acercar las capacidades de computación y almacenamiento al borde de la red, lo más cercano a los dispositivos de recursos limitados. Sin embargo, la computación *Edge* se enfoca principalmente en los dispositivos finales mientras que la computación *Fog* se centra en el lado de la infraestructura. La tabla 2.1 enumera las principales diferencias entre ambos tipos de paradigmas de computación.

- Paradigma de computación *Mist* [48–50]: la computación *Mist* se podría considerar como la versión más dispersa de la computación *Fog* ya que acerca el cómputo al extremo de la red llegando a involucrar incluso a los sensores y a los dispositivos actuadores. Esto disminuye notablemente la latencia, aumenta la autonomía de los subsistemas y la autoconciencia de cada dispositivos se vuelve crucial ya que su participación en el sistema depende completamente del entorno. Los principios que rigen las arquitecturas derivadas de este tipo de paradigma *Mist* son los siguientes cuatro:

  1. La red debe proporcionar información y no simplemente datos.

2. El uso de la red debe ser eficiente, es decir, solo debe entregar la información requerida cuando se la haya solicitado evitando mensajes *broadcast* o *multicast*.
3. Los dispositivos deben trabajar conjuntamente siguiendo un modelo de publicación-suscripción bajo las necesidades que requiera de manera dinámica el sistema en cada momento.
4. Los dispositivos deben ser sensibles al contexto en el que están desplegados y ser capaces de adaptarse a las necesidades de solicitud de información y configuraciones dinámicas de la red. No deben tener por tanto reglas configuradas de manera estática que les impidan descubrir dinámicamente los diferentes dispositivos o servicios disponibles dentro de su subsistema.

- Paradigma S-IoT (*Social Internet of Things*) [51, 52]: la idea básica de este enfoque consiste en la comunicación entre dispositivos conectados que permite las relaciones e interacciones de manera autónoma entre dichos dispositivos y las personas. Estas interacciones pueden ser también gestionadas por parte de las personas, las cuales pueden gestionar el comportamiento de cualquier dispositivo incluido en esta red. Para lograr cumplir la funciones que plantea este enfoque, las arquitecturas que se derivan de él se dividen en los siguientes componentes:
  - Gestor de relaciones (RM - *Relationship Management*): permite a los dispositivos iniciar, actualizar o terminar cualquier tipo de relación, existiendo para ello, un control humano en la gestión de las relaciones de amistad entre dispositivos.
  - Gestor de integridad (TM - *Trustworthiness Management*): este componente es el encargado de comprender cómo se procesa la información proporcionada por otros dispositivos estableciendo la confianza en dichos dispositivos en base a su comportamiento en el sistema.
  - Servicio de descubrimiento (SD - *Service Discovery*): es el servicio de búsqueda de relaciones dentro de la red. Actúa de manera similar a la búsqueda de relaciones e información por parte de las personas.

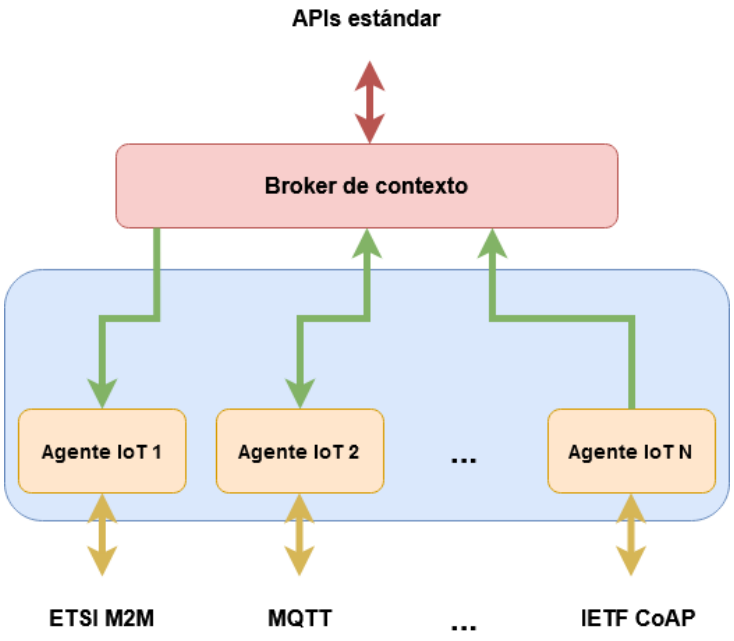

Figura 2.4: Arquitectura IoT-A (*Internet of Things - Agents*)

- Servicio de composición (SC - *Service Composition*): el descubrimiento de servicios utiliza las relaciones entre dispositivos para localizar el servicio requerido en cada momento, permitiendo así tanto un enfoque reactivo como proactivo en dicha composición del servicio. Dicho servicio incluye funciones de procesamiento de la información obtenida de los diferentes dispositivos para lograr aportar la respuesta más confiable a las consultas.
- Servicios (*Service APIs*): son los servicios que proporcionan la funcionalidad concreta al ecosistema y que consumen los diferentes dispositivos.

- Paradigma IoT-A (*Internet of Things - Agents*) [53]: este paradigma IoT-A se basa en componentes denominados agentes que son los encargados de lidiar con los aspectos de seguridad y comunicaciones que suponen una mayor carga para los dispositivos de recursos limitados. Esto permite a los

dispositivos que sus datos sean enviados y gestionados por un centralizador o *broker* de contexto usando sus propios protocolos nativos, los cuales son traducidos por estos agentes.

Generalmente, esta capa de agentes no está realmente embebida en los componentes del mundo físico por lo que si tomamos como referencia la imagen de la arquitectura mostrada en la figura 2.4, se puede observar como existen dos niveles de comunicación. El primer nivel se centra en abstraer la comunicación a bajo nivel con el dispositivo IoT, y el segundo, permite la comunicación entre los múltiples agentes y el *broker* de contexto. Esto permite un interfaz estándar para todas las interacciones de los dispositivos con el *broker* de contexto a través del patrón de diseño *facade* que se encarga de gestionar la complejidad de la heterogeneidad de los dispositivos.

2.2.4. Líneas de investigación y retos

Existen multitud de similitudes entre las características que definen el Internet de las cosas y el *cloud* como son, entre otras, la escalabilidad y el acceso ubicuo a los recursos. Sin embargo, estas características intrínsecas de los entornos *cloud* presentan muchos retos a la hora de ser abordadas por los dispositivos de recursos limitados que conforman el denominado Internet de las cosas. Entre los diferentes retos de base que se plantean encontramos [54]: permitir y facilitar el acceso y la conectividad al alto volumen de dispositivos heterogéneos que pueden encontrarse en movilidad a través de estándares y APIs, gestionar y orquestar millones de dispositivos y usuarios que generan cantidades masivas de datos, maximizar la utilización y compartición de recursos entre dispositivos, aplicaciones y plataformas, y la capacidad de personalización necesaria para cubrir los requerimientos de su contexto en el mundo físico.

Estas particularidades que convierten al Internet de las cosas en el hermano pequeño del *cloud* ya que los dispositivos que se engloban en el primero no pueden competir contra los recursos de los que disponen los servidores y aplicaciones desplegados en el segundo, fuerza a que surjan diferentes líneas de investigación en los aspectos clave que diferencian ambos mundos tan similares y a la vez tan distintos [55–57]:

- Estandarización: el mayor de los retos del Internet de las cosas es la propia adopción de estándares que permita una integración e interoperabilidad entre los distintos dispositivos que soportan diferentes tecnologías. La ausencia de una guía de estándares entre desarrolladores y corporaciones provoca la aparición de formas disruptivas de operar en Internet con vistas a abaratar diseños y configuraciones, lo que suele conllevar consecuencias indeseables para los recursos de red. Dentro de esta empresa titánica de intentar generar estándares de implementación tanto de hardware, software, elementos de red y tecnologías utilizadas, podemos encontrar grandes organizaciones como ETSI (*European Telecommunications Standards Institute*), ITU (*International Telecommunication Union*), IETF (*Internet Engineering Task Force*) o IEEE (*Institute of Electrical and Electronics Engineers*).

- Escalabilidad: aunque el reto de las arquitecturas escalables ya estaba llegando a un nivel de madurez bastante avanzado para los entornos *cloud*, ahora una arquitectura del Internet de las cosas debe ser capaz de incorporar nuevos dispositivos que en este caso disponen de recursos mucho más limitados que los de un servicio desplegado en el propio *cloud* y además, debe ser capaz de gestionar las diferentes tecnologías y comunicaciones de los dispositivos del Internet de las cosas, lo que vuelve a poner de manifiesto la necesidad de estudio de esta rama de conocimiento. Estas nuevas arquitecturas deben ser capaces de aportar fiabilidad, escalabilidad y confiabilidad en cualquier tipo de entorno, disponiendo además, de la capacidad de soportar múltiples dominios simultáneamente y de manera automatizada.

- Seguridad: la seguridad es el pilar esencial del Internet de las cosas y por tanto, otro de sus grandes retos. El crecimiento exponencial del número de dispositivos de bajo coste conectados y con bajos niveles de estandarización incrementa la probabilidad de robo de información que podría poner en riesgo la seguridad y salud de las personas. Los principales problemas de seguridad que se pueden encontrar en este contexto están relacionados principalmente con la IAAA (Identificación, Autenticación, Autorización, Auditoría), el direccionamiento, el nombrado, la confidencialidad en las comunicaciones, y las limitaciones de los propios recursos de los dispositivos.

Entre las diferentes limitaciones de los dispositivos se encuentran sus capacidades reducidas de batería o de cómputo, que dificultan, entre otros, el uso de métodos de autenticación robustos o que incluyan doble factor de acceso, o el uso de protocolos de cifrado en las comunicaciones o cifrado de los datos que se transmiten si para dicho cifrado se necesita una cantidad de cómputo por encima de sus capacidades. Naturalmente, todas estas condiciones complican también la capacidad de activar mecanismos tradicionales de seguridad como antivirus o herramientas de red como *firewalls*, y por tanto, esto permite a un atacante que al vulnerar un único dispositivo pueda llegar a comprometer todo el sistema.
Obviamente, las limitaciones de los dispositivos no sólo radican en características meramente de seguridad, sino que también, sus propias capacidades de comunicación mediante protocolos ligeros fuerza la existencia de algún tipo traductor de protocolos de comunicación entre los soportados por los dispositivos y por el *cloud*. Esto deriva en problemas de cómo llevar a cabo el direccionamiento entre el propio *cloud* y los dispositivos, y cómo realizar su nombrado y posible vinculación con la identificación de los mismos que puede no ser lo suficientemente robusta.

- Privacidad: las limitaciones de seguridad y robustez ya descritas con anterioridad se traducen en que cada nuevo dispositivo conectado tiene un potencial riesgo de ser atacado en cualquiera de los puntos de la red o sistema en el que se encuentre. Cuando a esto se suma un despliegue a gran escala en el que los dispositivos colaboran entre sí dentro de contextos de conocimiento colectivo, la información que dichos dispositivos pueden recolectar del entorno en el que están desplegados incluyendo información de los usuarios que se encuentran dentro de él, puede derivar que una fuga de información en dichos contextos contenga información considerada como sensible del propio usuario. Por este motivo, diferentes líneas de investigación se centran en evitar que esa información sensible del usuario se vea en peligro en los diferentes dispositivos en caso de que dicho dispositivo se vea comprometido, intentando preservar así, la privacidad de los usuarios.

- Regulación: dada la gran diversidad de aplicaciones del Internet de las cosas

y de las jurisdicciones bajo las cuales están desplegadas, existe un amplio grado de regulaciones y cuestiones legales sobre dicho ecosistema que plantea un reto en sí mismo. Algunos de los puntos que se intentan cubrir desde el ámbito regulatorio son, por ejemplo, la política de retención y destrucción de datos, la gestión de brechas de seguridad u otros. Sin embargo, la falta de una regulación global que permita definir un conjunto de reglas, procesos, protocolos o auditorías genera un vacío enorme dentro del sector del Internet de las cosas, que de ser cubierto, podría ser de extremada ayuda para las organizaciones desde un punto de vista de eficiencia y confiabilidad en sus sistemas reduciendo futuros errores.

La mayor parte de estos retos listados están siendo abordados mediante el uso del paradigma de computación *edge* combinado con la computación *cloud*. Gracias a los dispositivos *edge*, este enfoque permite que los recursos puedan organizarse dinámicamente gracias a las características de balanceo de carga, alta disponibilidad, conocimiento del contexto, proximidad con el mundo físico y calidad en el servicio (QoS - *Quality of Service*) [58] que proporcionan al sistema.

Las características de almacenamiento y cómputo proporcionadas por los dispositivos *edge* a los dispositivos de recursos limitados permiten la capacidad de almacenar y analizar en tiempo real la gran cantidad de datos generados o recopilados por el sistema mejorando la experiencia de usuario, reduciendo el tiempo de respuesta y optimizando el ancho de banda de la red.

Sin embargo, el *edge computing* genera también nuevos retos a resolver como son [59]: gestión remota de recursos, planificación de trabajo y distribución de dicha carga, tolerancia a fallos, gestión de respaldo de los datos, limitaciones de recursos y cómputo, autenticación, y diferentes aspectos relacionados con la privacidad y la seguridad.

2.3. Gestión de identidades y accesos en IoT

La gestión de la identificación, autenticación y autorización de los dispositivos englobados dentro del Internet de las cosas dadas sus características de heterogeneidad, escalabilidad, dinamismo y limitaciones tanto de recursos como capaci-

dades, es el gran reto a solucionar en este momento dentro de este ámbito. Más concretamente, la gestión de autorización y control de acceso a recursos y servicios con un nivel de permisos adecuado se convierte en un aspecto recurrente a solventar, ya que si se produce una identificación o autenticación precaria de un dispositivo de recursos limitados y finalmente resulta que dicho dispositivo ha sido suplantado por uno malicioso, se podría llegar a comprometer todo el sistema. Hay que tener en cuenta también que a mendo, en los procesos de autorización, es necesario la mediación por parte de los usuarios para autorizar o denegar el acceso de dichos dispositivos de recursos limitados a los recursos protegidos, lo que añade a la ecuación la interacción humana en un contexto diseñado para ser altamente escalable y dinámico.

En el momento de iniciar el análisis del estado del arte en el campo de la autorización dentro del ecosistema del Internet de las cosas, existen dos grupos o enfoques claramente diferenciados que intentan cubrir esta problemática.

En el primer enfoque se intenta proponer mecanismos de autorización simples y eficientes que no consuman todos los recursos disponibles de los dispositivos siguiendo una aproximación distribuida y cooperativa entre ellos. Para cumplir este objetivo, estos mecanismos se apoyan frecuentemente en capacidades criptográficas que se basan en características hardware de los propios dispositivos, protocolos de comunicación y de los dominios de aplicación [60–62]. En estos trabajos, el proceso de autorización descrito, aún existiendo interacción humana, es claramente eficiente y consume poco recursos de los dispositivos. Sin embargo, para poder disponer de un proceso de autorización así, es necesario introducir capacidades criptográficas extra dentro de los dispositivos, lo que implica que dichos dispositivos dispongan de características específicas que exceden de las necesidades de su dominio de aplicación y que son simplemente requeridas por criterios de seguridad, como por ejemplo: almacenamiento seguro para custodiar credenciales de acceso, una mayor capacidad de cómputo para poder ejecutar las distintas operaciones criptográficas requeridas, lo que implica un mayor consumo de batería, u otros. Por tanto, se considera que este enfoque está en contra de una de las principales características del ecosistema del Internet de las cosas; el ahorro de costes en el desarrollo de los dispositivos mediante el cual se minimiza la funcionalidad incluida en los dispositivos para que sean eficientes en el dominio de aplicación

concreto en el que se despliegan.

El segundo enfoque se basa en aproximaciones centralizadas o federadas que confían en alguna clase de motor o servidor de autorización central que incrementa la complejidad del proceso de autorización pero aporta una mayor maniobrabilidad y granularidad en la gestión de control de acceso basándose, entre otras características, en atributos del contexto tanto del dispositivo como de la sesión en la que se solicita el recurso en cuestión o incluso, permitiendo una autorización adaptativa basada en el riesgo identificado en la propia operación o solicitud. Por este motivo, este tipo de soluciones intentan delegar toda la carga de trabajo y complejidad de sus procesos de autorización en dicho motor centralizado, el cual, en muchas ocasiones se ejecuta en una máquina en el *cloud* sin casi limitación de recursos en comparación con los propios dispositivos IoT. Gracias a la existencia de dichos motores centrales de autorización, este enfoque puede aprovecharse de la potencia de los modelos de control de acceso más maduros y extendidos a día de hoy como son el control de acceso basado en roles (RBAC - *Role-Based Access Control*) y el control de acceso basado en atributos (ABAC - *Attribute-Based Access Control*), además de tener la opción de ampliar su abanico de posibilidades hacia modelos de control de acceso más vanguardistas y útiles dentro del contexto del Internet de las cosas como las soluciones federadas de control de acceso basado en *tokens* (*Token-based Access Control*) o incluso, haciendo uso de *blockchain* y contratos inteligentes (*smart contracts*). En la tabla 2.2 se puede observar una recopilación de los trabajos más significativos englobados bajo este enfoque y sus características particulares donde los términos abreviados utilizados dentro de dicha tabla son:

- RBAC = Control de acceso basado en roles (*Role-Based Access Control*).
- ABAC = Control de acceso basado en atributos (*Attribute-Based Access Control*).
- *Token* = Control de acceso basado en *token* (*Token-based Access Control*).
- Block. = Basado en blockchain o *smart contracts*.
- Crypt. = Requiere capacidades criptográficas en los propios dispositivos.
- Authn. = Proporciona autenticación.
- Del. = Soporta delegación de autorización entre dispositivos.

Tabla 2.2: Comparación de trabajos previos en gestión de identidades en el IoT

Ref.	RBAC	ABAC	*Token*	Block.	Crypt.	Authn.	Del.	Aut.	HTTP	Light.
[63]		X			X	X	X		X	
[64]		X						X		X
[65]	X				X	X		X		X
[66]			X					X		X
[67]		X			X	X			X	
[68]				X	X	X	X		X	
[69]			X					X		X
[70]		X				X				X
[71]			X		X	X			X	
[72]	X						X		X	
[73]				X	X	X			X	
[74]	X				X	X		X	X	
[75]	X							X		X

- Aut. = Soporta un registro o *enrol* de dispositivos automatizado y escalable.
- HTTP = Los dispositivos requieren soportar HTTP.
- Light. = Los dispositivos soportan protocolos ligeros como CoAP o MQTT.

En la recopilación de trabajos mostrados en la tabla 2.2 se percibe varios puntos interesantes dentro de este enfoque centralizado en los que existen diferentes aspectos de mejora a cubrir para poder adaptar un modelo de gestión de identidades y accesos a las diferentes particulares de los dispositivos del Internet de las cosas. Dichos puntos identificados son los siguientes:

1. Se observa que la mayor parte del estado del arte actual de este segundo enfoque fuerza a que si un dispositivo IoT quiere ser autenticado en el sistema para poder confiar en su identificación, éste debe poseer características criptográficas que den soporte, entre otros, a la custodia segura de las credenciales de autenticación del propio dispositivo. Este aspecto es un punto bastante limitante y provoca una gran similitud con el primer enfoque ya mencionado en esta sección.

2. Se identifica la existencia de cierta concordancia entre el uso de protocolos de comunicación ligeros y la capacidad de llevar a cabo, gracias a dichos protocolos, de una gestión de registro automático en el sistema de los diferentes dispositivos IoT. Sin embargo, y vinculado con este dinamismo, se aprecia que en caso de disponer de la capacidad de automatizar el proceso

de registro en el sistema, la gestión de autorización o de delegación de autorización entre dispositivos más robustos y los dispositivos IoT con recursos más limitados desaparece, es decir, una gestión de autorización compleja parece estar reñida con una gestión escalable y automática de registro en el sistema para dispositivos IoT.

3. Finalmente, y relacionado con los modelos de control de acceso, se puede reseñar que el uso de protocolos de comunicación ligeros y la automatización del flujo de registro de los dispositivos permite, en su mayor parte, llevar a cabo una gestión de autorización enriquecida que se apoya en los modelos de control de acceso basados en atributos o *tokens*, por lo que se puede concluir que éstos son los modelos de control de acceso más adecuados para ser utilizados en el ecosistema del Internet de las cosas.

Identificados los aspectos clave en base al estado del arte actual se llega a la conclusión de que es necesario cubrir los diferentes problemas de gestión de autorización y delegación de la misma entre dispositivos IoT permitiendo un registro automatizado de dispositivos IoT en el sistema con unos niveles de autenticación e identificación adecuados sin necesidad de encarecer la construcción del dispositivo añadiendo características criptográficas extra y soportando protocolos de comunicación ligeros.

2.4. Direccionamiento y nombrado en IoT

El direccionamiento y nombrado dentro del ámbito del Internet de las cosas plantea también un reto importante de la misma forma que la gestión de control de acceso a los recursos *cloud* por parte de los dispositivos IoT. Esto es así ya que al estar los dispositivos desplegados en el mundo físico, éstos necesitan poder comunicarse con el *cloud* y viceversa para poder cubrir las funciones necesarias del sistema concreto en el que están desplegados. Sin embargo, cabe destacar que aunque el direccionamiento y nombrado plantean un reto importante en dicho contexto, ambos suelen ser considerados como un medio y no como un fin en sí mismo, y de ahí que el estado del arte en dicha materia suela estar ligado a su vez a la resolución de otras limitaciones.

En este sentido, el enfoque más ampliamente utilizado a día de hoy en este campo se fundamenta en abordar el direccionamiento de una manera jerárquica entre los dispositivos englobados dentro del Internet de las cosas y los entornos *cloud*. Dentro de este enfoque existen dos vertientes:

1. El primer conjunto de trabajos dentro de este enfoque jerárquico fuerzan a que los servicios *cloud* tengan conocimiento de las características hardware y particulares técnicas de los dispositivos así como de los protocolos de comunicación ligeros que utilizan para interactuar con el propio *cloud*, lo que supone un fuerte acoplamiento del sistema concreto con el caso de uso o problemática a resolver [76, 77]. Estas características físicas de los propios dispositivos, en algunas ocasiones, no sirven sólo para enriquecer el direccionamiento y el nombrado entre ambos contextos sino que son determinantes para llevar a cabo dicho direccionamiento desde el entorno *cloud* [78, 79], lo que acopla aún más el entorno *cloud* a los detalles de implementación de los dispositivos, impidiendo así, soportar otra característica principal del IoT: la heterogeneidad.

2. Por otro lado, el segundo conjunto de trabajos intenta independizar el *cloud* de las características de bajo nivel de los dispositivos IoT manteniendo el direccionamiento jerárquico planteado con el fin de incrementar la interoperabilidad con los diferentes dispositivos heterogéneos. Para ello, esta vertiente se apoya en el uso de los dispositivos de borde o dispositivos *edge* [80, 81], lo que permite al sistema *cloud* abstraerse de los detalles técnicos de los dispositivos reduciendo así la complejidad que deben gestionar y delegando dicha complejidad en esta capa intermedia. La inclusión de estos dispositivos de borde en el medio de la jerarquía de direccionamiento entre el *cloud* y los dispositivos IoT permite que dichos dispositivos de borde actúen como *routers* NAT [82] (*Network Address Translation*) entre ambos contextos sirviendo, entre otros, de traductores de protocolos de comunicación. Obviamente, aunque esta vertiente es más interoperable e independiza al *cloud* de una conexión directa punto a punto con los dispositivos, esta nueva capa puede provocar, entre otros, inconsistencias en las traducciones de los mensajes de comunicación o en el aseguramiento de la recepción de

los mensajes por los dispositivos.

El direccionamiento jerárquico muestra una clara madurez a la hora de ser el más idóneo para ser utilizado en el contexto del Internet de las cosas. El único inconveniente que se puede observar en este primer enfoque descrito está relacionado con que el direccionamiento sigue el patrón maestro-esclavo, es decir, el *cloud* es el nodo maestro dentro del sistema y único responsable del correcto funcionamiento del direccionamiento y los diferentes dispositivos son los nodos esclavo que reciben las diferentes solicitudes u órdenes mediante el envío de peticiones síncronas punto a punto desde el *cloud*.

Dadas las características de escalabilidad de los ecosistemas del Internet de las cosas, surge un segundo enfoque con el fin de delegar la responsabilidad entre los diferentes dispositivos para no sobrecargar de gestión el *cloud*. Este segundo enfoque se apoya en el patrón de publicación-suscripción que proporcionan las arquitecturas orientadas a eventos (EDA - *Event-Driven Architecture*) [83, 84]. Este tipo de arquitecturas, como se muestra en la figura 2.5, define un conjunto de roles claramente diferenciados que le permite soportar una alta escalabilidad dentro del sistema donde se implante en comparativa con las arquitecturas tradicionales de peticiones síncronas punto a punto:

- Colector de eventos: es la pieza central en base a la cual gira este tipo de arquitecturas. Dependiendo de la implementación, el colector de eventos puede disponer de persistencia en disco o estar ubicado en memoria volátil. En contraposición del envío punto a punto entre los distintos nodos del sistema, el colector de eventos, como intermediario, recibiría las peticiones del emisor para que posteriormente, uno o múltiples nodos receptores del mensaje, soliciten a este colector de eventos dicho mensaje para iniciar su procesamiento. La asincronía por tanto se consigue gracias a que el nodo emisor considera que su petición ha sido correcta cuando el colector de eventos le confirma que ha almacenado el mensaje para futuras solicitudes de los receptores, aunque dichas solicitudes no realicen el procesamiento completo requerido por el nodo emisor del mensaje hasta un tiempo después cuando finalmente obtengan dicho mensaje del colector de eventos.

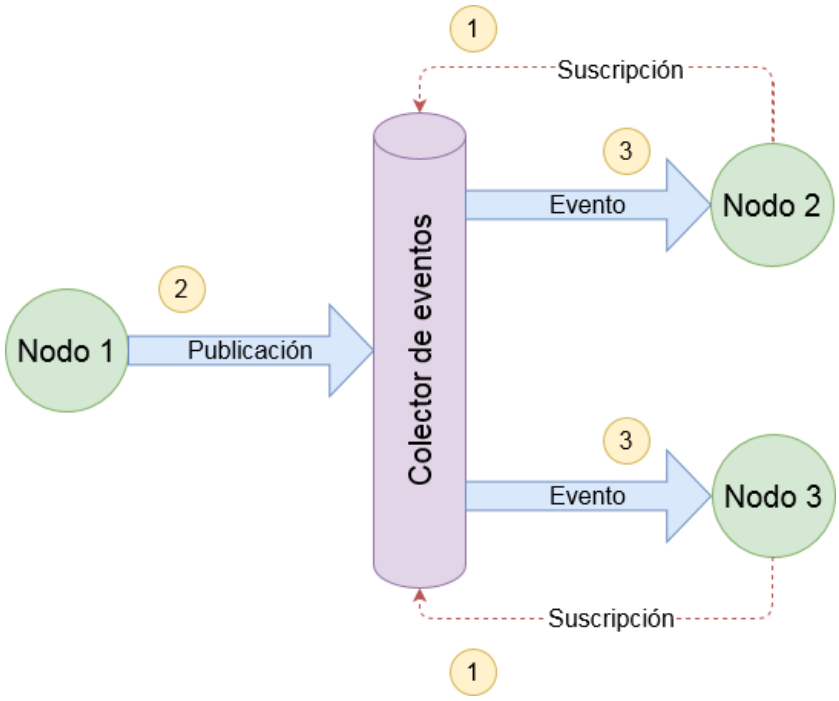

Figura 2.5: Arquitectura orientadas a eventos

- Nodos: los nodos que conforman el sistema pueden cubrir ambas funciones, es decir, ser productores de mensajes o suscriptores de los mismos. Esta dualidad permite a cualquier nodo del sistema suscribirse a un colector de eventos concreto para que cuando un nodo emisor publique un mensaje, todos los nodos suscritos a dicho colector puedan recibirlo e iniciar su procesamiento. Como son los propios nodos los que llevan a cabo su registro y suscripción al colector de eventos en vez de ser necesario un servidor de nombres que resuelva dónde está cada uno de ellos en cada momento y su disponibilidad, este tipo de arquitecturas permiten una alta escalabilidad en el número de nodos que los componen.

Los trabajos de este segundo enfoque que se apoyan en las características de las arquitecturas orientadas a eventos permiten una mayor escalabilidad, lo cual es un aspecto muy importante y beneficioso para el contexto del Internet de las cosas. Sin embargo, es la necesidad de suscripción por parte de los propios dispositivos IoT al colector de eventos sin ningún tipo de nodo intermedio lo que fuerza que

dichos dispositivos estén provistos de características criptográficas extra con el fin de soportar, entre otros, la custodia segura de sus credenciales de autenticación. provocando un mayor consumo de recursos y batería por parte de los mismos y confiriendo a este segundo enfoque, su inconveniente más notorio.

Por este motivo, y dado el estado del arte actual del direccionamiento, se llega a la conclusión de que es necesario cubrir las limitaciones de los modelos de direccionamiento existentes basándose para ello en un direccionamiento jerárquico tradicional que no fuerce a los dispositivos a tener que poseer características especifícas de seguridad, que abstraiga a los entornos *cloud* de cierta complejidad gracias a los dispositivos de borde o *edge*, y que incluya las ventajas de escalabilidad que proporcionan los direccionamientos basados en las arquitecturas orientadas a eventos.

Finalmente, y una vez identificado cómo se debe cubrir el problema del direccionamiento dentro de un sistema englobado en el Internet de las cosas, el siguiente reto es el de definir un nombrado unívoco de cada uno de los diferentes dispositivos que pueden ser accedidos gracias al modelo de direccionamiento en cuestión. El nombrado de un objeto no deja de ser una abstracción a alto nivel de la dirección física como puede ser por ejemplo, una dirección IP. Sin embargo, dicho nombrado permite, entre otros, una mejor gestión de registro de dispositivos en el sistema, capacidades de descubrimiento y definición de interrelaciones entre ellos, u otras características que permiten abstraer a la lógica de negocio del sistema de las diferentes características del direccionamiento y mapas de red utilizados en él con el fin de proporcionar una versión holística centrada en el dominio de aplicación y no en los detalles de implementación y despliegue. Con el fin de cubrir este objetivo, existen dos enfoques principales que intentan abordar la problemática que plantea el nombrado el contexto del Internet de las cosas:

1. En el primer enfoque se incluye, en el propio nombre otorgado a un dispositivo, información relativa a qué acciones admite o puede realizar dicho dispositivo [85–87]. Por ejemplo, en el caso de disponer de bombillas inteligentes, dentro del nombrado de cada una aparecerá que soporta la funcionalidad tanto de encenderse como apagarse, lo que provoca que en el caso de que todo el parque de bombillas no sea homogéneo, bombillas con

otras funcionalidades como parpadear o cambiar de color queden fuera del esquema de nombrado o, en el peor de los casos, sean consideradas como dispositivos independientes aunque estén desplegados en el mismo entorno que las primeras. Esta particularidad obviamente condiciona el esquema de nombrado a los detalles de implementación de los distintos dispositivos, lo que provoca que la heterogeneidad de dispositivos que pueden existir en el sistema de manera simultánea penalice la gestión de dicho esquema.

2. Por otro lado, los trabajos englobados dentro del segundo enfoque intentan abordar el esquema de nombrado de una forma completamente independiente de las propiedades de los dispositivos y de los detalles de implementación o funcionalidades que soportan [88, 89]. En dichos trabajos, se soporta por defecto el aspecto de la heterogeneidad de los dispositivos ya que al abstraerse de los detalles de implementación, se pone foco principalmente en las funcionales propias de los dominios de aplicación concretos en los que se despliega el esquema de nombrado. Esto les otorga una mayor aplicabilidad pero como contrapartida al primer enfoque, el dotar de una capa de abstracción al sistema ocultando las particularidades técnicas de los dispositivos desplegados provoca que el rendimiento de este tipo de sistemas sea menor ya que es necesario, a partir del nombre de un objeto, buscar en un registro independiente del nombre la funcionalidad soportada por dicho objeto para poder equipararse a las funcionalidades ofrecidas por el enfoque anterior.

Para concluir, y sumado a la conclusión del direccionamiento ya planteada donde la heterogeneidad de los dispositivos en el sistema es en muchos casos una necesidad, es necesario poder aplicar un esquema de nombrado adecuado para dicho contexto heterogéneo sin que el rendimiento se vea afectado por añadir la capa abstracción necesaria al sistema.

2.5. Conocimiento colectivo en IoT

La ingente cantidad de dispositivos desplegados dentro del mundo físico caracterizados por disponer de una múltiple variedad de sensores y de capacidades

de cómputo y comunicación, son capaces de proporcionar una enorme cantidad de datos y de información sobre el entorno concreto del que forman parte. La escalabilidad que tienen estos sistemas con el fin de obtener más y más información del entorno invita a querer hacer uso de esta capacidad de obtención de conocimiento distribuido como una forma de inteligencia colectiva. Este tipo de paradigmas basado en el conocimiento colectivo persigue la obtención de información del gran número de las partes que lo componen, ya sean dispositivos o personas, con el fin de analizar y utilizar dicha información obtenida en el beneficio de dicho compendio o multitud de partes en sí. Los diferentes puntos de vista proporcionados por la gran multitud de observaciones del entorno proporcionan al sistema en su conjunto la potencia necesaria para realizar diferentes tipos de agregaciones y análisis de los datos recopilados y tomar así, las mejores decisiones basadas en dichos datos.

Existen dos grandes paradigmas basados en el conocimiento colectivo que son las bases sobre las que descansan los distintos dominios de aplicación que utilizan estos modelos. Dichos paradigmas son [90, 91]:

- *Crowdsourcing*: es un tipo de paradigma participativo en el cual cada individuo o grupo de individuos de diverso nivel de conocimiento y capacidades colaboran de manera voluntaria con el fin de cumplir o realizar una tarea de interés común para ellos. La clase de colaboración de cada uno de los individuos, los cuales son principalmente personas, puede ser de diferente índole, es decir, cada uno de ellos puede participar en la realización de la tarea aportando parte de su trabajo, conocimiento o experiencia, o incluso, a nivel económico. Una vez completada la tarea, cada individuo obtendrá la contrapartida de su colaboración en función de la necesidad de satisfacción que tenga, pudiendo ser, una satisfacción económica, de reconocimiento social u otros. El ejemplo más representativo que ayuda a entender este tipo de paradigma colaborativo basado en la inteligencia colectiva de los individuos sería la Wikipedia.

- *Crowdsensing*: otro tipo de paradigma basado en el conocimiento colectivo está relacionado no tanto con la ejecución de una tarea llevada a cabo de manera colaborativa por cada una de las partes hasta la obtención de un resultado que beneficie al bien común, sino a la obtención de información

de diferentes sensores o incluso apoyándose en la percepción humana con el fin de ayudar a tener un conjunto de información tan rica, que permite al motor central del sistema la capacidad de tomar las mejores decisiones en base a los datos recabados por las diferentes partes. Uno de los métodos más utilizados a día de hoy que aprovecha este paradigma sería el basado en el uso y explotación de los dispositivos móviles, el cual se denomina *Mobile Crowdsensing* (MCS). Dicho modelo se sustenta sobre la base de que los dispositivos móviles disponen a día de hoy de una gran cantidad de sensores como acelerómetro, giroscopio, cámara, micrófono o GPS entre otros, que permiten observar el entorno de una manera sencilla gracias simplemente a la presencia de dicho dispositivo dentro del entorno a estudiar. A su vez, este tipo de paradigma de *crowdsensing* se subdivide en dos categorías cuya característica principal es el nivel de interacción humana en la captación de los diferentes datos del entorno:

- *Crowdsensing* participativo (en inglés, *Participatory crowdsensing*): este modelo participativo implica que el individuo esté activamente involucrado en la obtención de información a través de su dispositivo móvil. Con el fin de cumplir este objetivo, el propietario del dispositivo dispone de una aplicación concreta dentro del mismo con interfaz de usuario que le permite activamente realizar los reportes de las observaciones realizadas.
- *Crowdsensing* oportunista (en inglés, *Opportunistic crowdsensing*): este modelo de obtención de información se basa en un esquema de captación desatendido que suele materializarse mediante aplicaciones o procesos en segundo plano dentro del dispositivo móvil, el cual realiza la recopilación de la información y la envía al sistema de procesamiento central del dominio de aplicación concreto en el que está registrado ocurriendo todo ello, de una manera transparente y sin interacción por parte del individuo propietario del dispositivo.

Para el caso concreto del paradigma *crowdsensing* existen multitud de dominios de aplicación en los cuales se está llevando a cabo su implantación. Por

ejemplo, en campos como el sanitario, la obtención de datos apoyados en este paradigma incrementa la monitorización con el fin de ayudar a mejorar la capacidad diagnóstica. En otros como el ambiental o el de infraestructuras públicas, se puede monitorizar, entre otros, el nivel de polución y de hábitos salvajes, y cómo estos influyen en las condiciones de las diferentes carreteras. En el ámbito de las ciudades inteligentes también se están llevando a cabo diferentes casos de uso como son la monitorización de los atascos de tráfico o la disponibilidad en los aparcamientos. A continuación, se abordan los ejemplos más representativos dentro de los diferentes campos en los que a día de hoy se está aplicando el paradigma *crowdsensing* en cualquiera de sus categorías, es decir, participativa u oportunista:

- Sanidad: dada la naturaleza de la información que se maneja en estos sistemas y de los grandes beneficios que puede proporcionar para la calidad de vida humana, la monitorización por parte de un médico en remoto de diferentes parámetros de salud de un paciente gracias a diferentes sensores que proporcionan datos de manera desatendida como la saturación de oxígeno en sangre, las pulsaciones por minuto, la temperatura, el electrocardiograma, u otros, ha provocado que este campo sea uno de los más explorados [92–96]. La pandemia del COVID-19 ha incrementado la necesidad de impulsar aún más este ámbito con el fin de que los sistemas médicos se transformen en servicios en línea para poder llegar a los diferentes hogares aun existiendo las diferentes restricciones sanitarias definidas por los gobiernos de cada país [97]. Finalmente, esta pandemia también ha impulsado el nacimiento de multitud de aplicaciones de autodiagnóstico y reporte de casos de COVID-19 para tener una visión holística de la expansión de la pandemia en las diferentes geografías en base a la participación activa de la población [98–100].

- Medio ambiente: el rápido crecimiento demográfico y los efectos negativos que este está teniendo sobre el medio ambiente debido entre otros a la sobreexplotación de la agricultura y a la industrialización, ha provocado un deterioro, por ejemplo, en la calidad del aire debido a la emisión de diferentes tipos de gases de efecto invernadero a la atmósfera o afectado incluso a la calidad del agua. Por este motivo, dentro de este ámbito y con el fin de

monitorizar el medio ambiente no sólo desde el punto de vista de sensores físicos que permitan medir la temperatura y la humedad, también se están desplegando otro tipo de sensores químicos y biológicos que permiten obtener información sobre gases dañinos como monóxido de carbono, metano, hidrógeno, gases inflamables, bacterias en el agua, y un largo etcétera. Estos trabajos buscan llevar a cabo una obtención de información desatendida basada en estos sensores con el fin de monitorizar algo tan importante para la vida como es el medio ambiente y poder disponer así de datos sobre su deterioro [101–106], para el que si no se toman medidas, puede desembocar en futuros problemas para la salud del ser humano.

- Ciudades inteligentes: el objetivo principal del nacimiento de este ámbito está relacionado con la necesidad de incrementar la calidad de vida y comodidades de los residentes dentro de una ciudad. Para ello, se obtiene información de interés para los diferentes ciudadanos de tal forma que se les pueda facilitar de una forma sencilla de entender y fácil de utilizar. En este ámbito existe una colaboración entre las diferentes instituciones del gobierno que dotan de sensores y de capacidades a las propias ciudades, y la sociedad en sí, la cual puebla el sistema con información de interés derivada de su vida diaria como su movilidad, comportamientos económicos o de su entorno. Este tipo de compartición de información entre sociedad e instituciones gubernamentales plantea cuestiones relacionadas con el anonimato y la privacidad que por ejemplo en los casos anteriores, al no existir dos entes con objetivos distintos dentro del sistema, no se planteaban. Algunos de los ejemplos más significativos en los que se viene trabajando dentro de este ámbito incluirían la monitorización de la contaminación sonora [107–110], atascos de tráfico en el área metropolitana de la ciudad inteligente [111, 112], incidencias y emergencias [113], e incluso, detección de terremotos [114].

- Infraestructuras públicas: el ámbito de las infraestructuras públicas (o privadas según el caso) comprende el conjunto de instalaciones esenciales como carreteras, redes de suministro de agua, electricidad o telecomunicaciones, puentes, edificios y el resto de estructuras que dan soporte a las diferentes

necesidades de una ciudad o país. Dada la importancia de estas infraestructuras, se convierte en necesario desplegar un conjunto de componentes que permitan mantener y mejorar las condiciones y el estado de dicha infraestructuras con el fin de seguir dando servicio a las personas en su vida diaria. Un campo también estudiado dentro de este ámbito de las infraestructuras es el de la monitorización de tráfico para poder identificar los distintos comportamientos al volante de los conductores, incidencias repentinas en el tráfico, agresividad al volante, y por supuesto, estado del tráfico como atascos, cálculos de tiempo de viaje u otros [115–117]. Dicha monitorización del tráfico a su vez permite explorar y monitorizar las diferentes anomalías en el estado de las carreteras, ya que puntos propensos a accidentes o a atascos pueden ser debidos a un estado deficiente del pavimento, mala señalización u otras características que requieren una intervención por parte de las autoridades pertinentes para solventar dichas incidencias [118–120]. Finalmente, dentro de este ámbito también encontramos los sistemas de monitorización de salud de las estructuras en los cuales los sensores se encargan de detectar y localizar diferente tipología de daños estructurales que haya podido sufrir la estructura. Este tipo de proyectos están muy ligados a la ingeniería civil y por tanto, el uso de nuevas tecnologías para monitorizar las estructuras permiten detectar y prevenir daños en este tipo de infraestructuras [121–123].

Esta transformación de nuestro mundo hacia un ecosistema poblado de sensores que recopilan datos de manera masiva, lo ha convertido en un entorno observable y por tanto, medible. Sin embargo, esta recopilación de información de nuestro alrededor genera multitud de nuevos retos desde el punto de vista de la privacidad, ya que principalmente, nuestra identidad como usuarios del mundo físico y nuestra localización podría verse expuesta a terceras partes [124]. Teniendo en cuenta que gran parte de los trabajos realizados hasta la fecha en el campo del *crowdsensing* éstos se apoyan principalmente en la gran cantidad de capacidades que proporciona la vertiente del *mobile crowdsensing*, los problemas de privacidad se agravan aún más al ser el dispositivo del propio usuario la fuente de recopilación de dicha información.

La privacidad depende del escenario observado en cuestión, ya que no es lo

mismo el estudio del medio ambiente que un caso de estudio concreto con información sanitaria. Por este motivo, la preservación de la privacidad se debe enfocar como un punto importante e independiente de cualquier otro criterio de diseño, y por ende, debe ser un parámetro de suma importancia a la hora de llevar a cabo el diseño de la solución planteada. En este aspecto, muchos son los mecanismos disponibles que se pueden aplicar a la hora de diseñar un sistema que desee preservar la privacidad dentro de contextos de *crowdsensing* en función del dominio de aplicación concreto. Dichos mecanismos se pueden dividir en los siguientes grupos [125, 126]:

- Mecanismos para tareas de procesamiento: las tareas de procesamiento son aquellas que en tiempo real llevan a cabo una serie de cálculos de agregación, estimación o manipulación sobre los datos recabados con el fin de disponer de unos datos de salida que permitan tomar decisiones en caliente para modificar el comportamiento del sistema en base a las observaciones realizadas del entorno, y por tanto, influenciar en el mundo físico de una manera adecuada. Al ser datos obtenidos y procesados en tiempo real, las preocupaciones sobre los aspectos de privacidad están muy ligadas a la identidad del usuario y su ubicación. Por este motivo, dichas preocupaciones se intentan abordar mediante el uso de una autenticación anónima o basada en atributos que permita saber que el dispositivo es uno de los permitidos en el sistema sin exponer la identidad del usuario.

- Mecanismos para tareas de reporte: las tareas de reporte no están tan ligadas con la interacción en tiempo real con el mundo físico sino que suelen apoyarse en una mayor capacidad de cómputo de infraestructuras como el *cloud* con el fin de obtener información agregada de comportamientos y tendencias de la multitud, o realizar predicciones futuras en base a la gran cantidad de datos recopilados por cada uno de los usuarios que conforman dicha multitud. En este aspecto, se puede utilizar dos enfoques a la hora de custodiar y procesar dicha información masiva de manera que se preserve la privacidad de los usuarios. El primer enfoque consiste en la propia modificación de los atributos recopilados mediante diferentes técnicas de anonimización y ofuscación de los datos para que no sea posible hacer coincidir

Tabla 2.3: Comparación de trabajos previos en privacidad en IoT para esquemas *crowdsensing*

Referencia	*Cloud*	*Edge*	Criptografía	Anonimización	Ofuscación
[127]		X			X
[128]		X	X		
[129]		X	X		
[130]		X			X
[131]	X			X	
[132]		X			X
[133]		X			X
[134]	X		X		
[135]	X		X	X	

los datos recabados con la identidad del usuario que los proporcionó. El otro enfoque, al contrario del primero, no realiza ningún tipo de modificación de los diferentes atributos, pudiendo hacer uso para ello de técnicas de cifrado o simplemente, activando y desactivando la capacidad de recopilación de información cuando el usuario desee hacerlo de manera voluntaria.

Estas técnicas están siendo ampliamente utilizadas a día de hoy, como ya se ha comentado, para tratar los diferentes problemas de privacidad que se plantean en los esquemas de *crowdsensing* y *mobile crowdsensing* como se muestra en la tabla comparativa 2.3, donde los términos abreviados utilizados en ella son:

- *Cloud* = La solución propuesta sigue un enfoque *cloud-based*.
- *Edge* = La solución propuesta se apoya en arquitecturas *edge*.
- Criptografía = Utiliza técnicas criptográficas.
- Anonimización = Utiliza técnicas de anonimización.
- Ofuscación = Utiliza técnicas de ofuscación.

En dicha tabla se puede identificar como los enfoques en los que se utilizan unas técnicas criptográficas computacionalmente más costosas como los cifrados homomórficos siguen una aproximación *cloud* y no suelen delegar sus responsabilidades en dispositivos *edge*. Sin embargo, existe una tendencia creciente en el uso de arquitecturas *edge* dentro de estos esquemas al requerirse cada vez más un comportamiento en tiempo real y donde se utilizan técnicas menos costosas computacionalmente como la ofuscación de los datos.

La necesidad principal del uso de este tipo de técnicas para proteger la privacidad del usuario viene derivada del reto que plantea el *crowdsensing* en sí: el incentivo o recompensa económica al usuario para que sacrifique parte de su privacidad en beneficio del enriquecimiento de información de calidad disponible para el sistema. Obviamente, ningún usuario sacrificaría su privacidad sin la existencia de algún tipo de incentivo o recompensa económica que le compense, y por tanto, parece ser necesario disponer de la capacidad de gestionar micropagos dentro de las plataformas de *crowdsensing* para atraer a dichos usuarios.

La existencia de estos micropagos son los responsables de que los datos recabados por los diferentes sensores tengan que estar vinculados a una identidad para poder valorar su calidad, y posteriormente, poder recompensar en consecuencia al usuario, siendo normalmente mejor recompensado el usuario que sacrifica más su privacidad para proporcionar datos más enriquecidos el sistema. Esto plantea por tanto el reto de cómo se lleva a cabo la asignación de las tareas remuneradas entre los diferentes usuarios, existiendo dos enfoques principales [136]: en el primero es la propia plataforma la encargada de repartir las diferentes tareas entre los usuarios disponibles, y en el segundo, son los propios usuarios los que mediante un sistema de subastas ofrecen sus servicios al sistema y es éste quien deriva a posteriori las tareas al mejor postor.

Por este motivo, y dado el estado del arte actual de la privacidad en IoT en los esquemas de *crowdsensing*, se llega a la conclusión de que aprovechandose de las arquitecturas *edge* y la gran multitud de sensores desplegados en el mundo físico a día de hoy, es necesario revisar la necesidad de incentivación de los usuarios en dichos esquemas, ya que dicha incentivación provoca la mayor parte de los problemas de privacidad encontrados en los trabajos actuales y lastra la información disponible en los diferentes sistemas y dominios de aplicación.

Capítulo 3

Esquema de delegación de autorización para dispositivos IoT

El presente capítulo desarrolla la gestión de identidades planteada en esta tesis para dispositivos englobados dentro del Internet de las cosas (IoT). Dentro de dicha gestión se cubre el registro en el sistema de dichos dispositivos de recursos limitados, la delegación de autorización de los dispositivos de borde (o también llamados dispositivos *edge*) sobre éstos para que puedan acceder a los recursos protegidos ubicados en el *cloud*, y finalmente, el direccionamiento y el nombrado necesario para que los servicios *cloud* puedan mandar órdenes o modificar el estado de los dispositivos IoT en función de las decisiones que se consideren necesarias o mejores para el sistema.

Antes de abordar la definición del diseño de la delegación de autorización de dispositivos *edge* a dispositivos IoT, es necesario sentar las bases sobre las que se fundamenta el trabajo realizado en esta tesis así como la motivación del mismo. Por este motivo, en primera instancia se abordará, por un lado, la motivación y caso de uso que ha provocado la definición planteada, y por otro, la arquitectura base tomada de referencia dentro del ecosistema IoT así como las asunciones y tecnologías elegidas que delimitan de un modo u otro los casos de uso de aplicación que encajan en el modelo planteado.

3.1. Motivación

El alto grado de escalabilidad y heterogeneidad de un ecosistema IoT requiere de un nivel de automatización en la gestión de recursos superior al que se puede encontrar en el resto de ámbitos de las tecnologías de la información y las comunicaciones. Esto es así ya que este tipo de ecosistemas disponen de numerosos dispositivos con recursos y capacidades limitadas que utilizan protocolos de comunicación ligeros y que no son capaces de custodiar de manera segura ningún tipo de credencial o *token* de acceso. Aún con estas limitaciones, los diferentes dominios de aplicación despliegan una gran cantidad de estos dispositivos y por tanto, se hace necesario plantear cómo resolver los retos más importantes a los que se enfrenta un sistema cuando se incluyen estos dispositivos de recursos limitados en él sin comprometer el sistema en su conjunto como son: la identificación, el direccionamiento y el nombrado, la autenticación y autorización. Buenos ejemplos de dominios de aplicación en los que encontramos esta casuística concreta serían las ciudades inteligentes, la industria 4.0, la agricultura inteligente o incluso, diferentes escenarios sanitarios.

Las limitaciones y aspectos en común que encontramos en cada uno de los diferentes casos de uso mencionados que intentan abordar los trabajos analizados en el capítulo 2 de estado del arte se pueden resumir en los siguientes puntos:

- La autorización basada en *tokens* de acceso (TBAC - *Token-Based Access Control*) resulta ser mucho más eficiente que los enfoques tradicionales como RBAC o ABAC, los cuales habitualmente están más acoplados a las características específicas de los dispositivos IoT o de los protocolos de comunicación subyacente que utilizan. Este aspecto justifica el porqué el estado del arte de los modelos de control de acceso está tendiendo hacia este tipo de soluciones ya que dichas soluciones basadas en *tokens* suelen ser más interoperables con el resto de entornos a través de las diferentes capas de red.

- Una alternativa interesante que gran cantidad de trabajos están utilizando a la hora de materializar sus enfoques de control de acceso basado en *tokens* es el uso del protocolo OAuth 2.0. Aunque dicho protocolo es una buena

alternativa para abordar esta problemática del control de acceso, presenta la limitación de que requiere de comunicaciones HTTP seguras sobre TLS, lo que no encaja con las propias capacidades de red disponibles en la mayor parte de dispositivos IoT. Además, dichos dispostivos precisan en su lugar del uso de protocolos de comunicación ligeros a nivel de aplicación y con otro tipo de canales de comunicación segura distintos a TLS que requieran de una menor capacidad de cómputo pero que proporcionen una confiabilidad similar de custodia de los mensajes en tránsito.

- Por otro lado, aunque los diferentes trabajos de investigación previos intentan adaptar OAuth 2.0 a su escenario o dominio de aplicación concreto para aprovecharse de este mecanismo de autorización basado en *tokens*, ninguno de ellos aborda el problema principal de la escalabilidad de dichos ecosistemas, es decir, cómo una gran cantidad de dispositivos IoT se registran en el sistema. En la mayor parte de los trabajos, la compartición de las credenciales de OAuth 2.0 a los diferentes dispositivos IoT se presupone que se realiza de manera manual y antes de iniciar cualquier flujo, lo cual obviamente, limita los niveles de escalabilidad de manera considerable.

- Las diferentes propuestas de autorización hasta la fecha no abstraen suficientemente a los servicios *cloud* de los detalles de implementación y despliegue relacionados con los dispositivos IoT. Además, al ser servicios *cloud*, éstos no proporcionan de por sí mecanismos de autenticación ligeros por lo que se ven forzados a integrarse con sistemas de autenticación externos para dar soporte a dicha funcionalidad. Sin embargo, este tipo de soluciones tampoco funcionan con la totalidad de los dispositivos IoT de capacidades reducidas ya que requiere que dichos dispositivos puedan custodiar de manera segura su propias credenciales de acceso o dispongan de capacidades criptográficas robustas.

- Con respecto al direccionamiento en este tipo de sistemas, se observa que las aproximaciones asíncronas basadas en patrones de publicación-suscripción resultan más eficientes que el direccionamiento tradicional altamente acoplado a las características de los dispositivos y a los protocolos de comu-

nicación subyacentes. Sin embargo, como son los propios dispositivos IoT los que se tienen que suscribir al bus de eventos para obtener las acciones u órdenes a realizar que le solicitan los servicios *cloud*, provoca que dichos dispositivos requieran de un mayor consumo de sus recursos al necesitar autenticarse por sí mismos a dicho bus de eventos. Hay que tener en cuenta que esta suscripción directa por parte de dispositivos de características tan limitadas, reduce sustancialmente los niveles de seguridad de los sistemas, los cuales en muchas ocasiones pueden llegar a ser críticos, ya que se expone el bus, que es la pieza central que contiene los eventos del sistema con la única limitación de seguridad de requerir una autenticación ligera que pueda ser soportada hasta por dispositivos IoT de capacidades y recursos limitados.

- Finalmente, los esquemas de nombrado sobre los que se apoyan estos ecosistemas, del mismo modo que como sucede con los modelos de autorización ya mencionados, no abstraen suficientemente a los servicios *cloud* de las características de bajo nivel de los dispositivos, ya que en muchas ocasiones, dichos esquemas de nombrado se basan en las direcciones lógicas de los dispositivos IoT o en las propias funcionalidades que éstos proporciona, lo cual evita que el esquema de nombrado sea interoperable entre distintos dominios de aplicación.

A todo esto hay que sumarle que, en la mayoría de despliegues, los dispositivos IoT son ubicados tras un *router* NAT [82] (*Network Address Translation*). Dichos *router* traducen una IP de un espacio de direcciones a otra modificando información específica de las cabeceras IP de los paquetes de red, lo cual permite a este router intermediario enmascarar tras una única dirección IP pública, una gran cantidad de direcciones de red privadas que no tienen porqué ser necesariamente IP. Esto ayuda a evitar la saturación del espacio de red de IPv4, que con la gran cantidad de dispositivos IoT que se pueden desplegar dentro de un sistema de manera escalable, provocaría el colapso de IPv4.

Por tanto, una vez tratados los aspectos y limitaciones comunes encontrados en los trabajos anteriores, el objetivo principal planteado a lo largo del trabajo de esta tesis es abordar dichas limitaciones adaptando OAuth 2.0 a los contextos

IoT apoyándose para ello en los dispositivos *edge* o de borde que se describirán en la siguiente sección de este capítulo de arquitectura de referencia. Este punto permite utilizar dichos dispositivos de borde con un enfoque similar al de los *router* NAT permitiendo delegar su autorización OAuth 2.0 a los dispositivos IoT de recursos limitados que apantalla y que no son capaces de comunicarse directamente con los servicios *cloud* a través de HTTP. Además, dicha adaptación cubre de manera completamente automatizada el proceso de registro en el sistema de los diferentes dispositivos con el objetivo de garantizar los niveles requeridos de escalabilidad que estos ecosistemas del Internet de las cosas requieren. Finalmente, y continuando con la aproximación centrada en los dispositivos *edge*, también se abordan las problemáticas del direccionamiento y nombrado de los dispositivos IoT en el sistema, absorbiendo las mejores características que proporcionan los enfoques basados en patrones de publicación-suscripción sin reducir las características de seguridad del sistema en su conjunto por haber incluido este tipo de direccionamiento.

3.2. Arquitectura de referencia y asunciones

Como se puede observar en la figura 3.1, el trabajo llevado a cabo en esta tesis sigue una aproximación centrada en los dispositivos *edge* o dispositivos de borde para abordar los problemas de autorización y control de acceso, direccionamiento, y nombrado de los dispositivos IoT. Este enfoque consiste en plataformas distribuidas desplegadas en los límites de las redes que sirven de puente entre el mundo físico cubierto por los dispositivos IoT, los cuales son las fuentes principales de datos del sistema que suelen estar altamente limitados desde el punto de vista de capacidades y recursos, y los propios servicios ofrecidos desde los entornos *cloud* a través de Internet. Por este motivo, la arquitectura tomada de referencia se fundamenta en los siguientes tres roles que cubren las tres capas planteadas:

- Servicios *cloud*: estos servicios, recursos o aplicaciones proporcionan, entre otras, capacidades de cómputo, comunicaciones, almacenamiento, etc. al sistema del que forman parte. Dentro del mundo del Internet de las cosas, estas entidades necesitan autorizar de algún modo a dichos dispositivos para

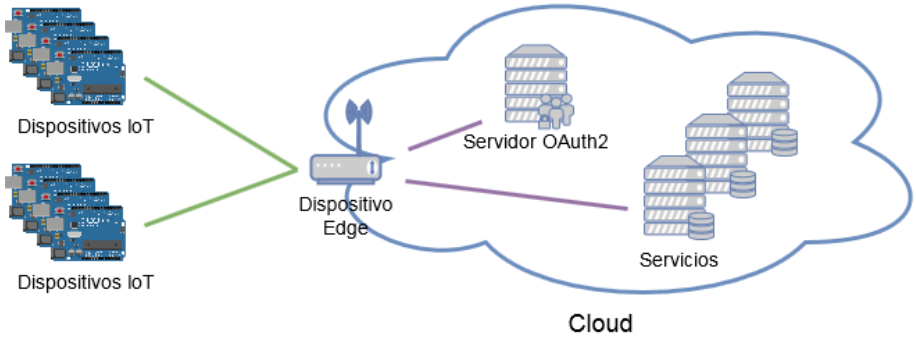

Figura 3.1: Arquitectura *edge-centric* de referencia

que lleven a cabo ciertas tareas sobre los recursos que ellos disponen en el *cloud*, ya que en la mayoría de los casos, los dispositivos IoT no disponen de tantas capacidades y recursos como un servicio *cloud*. En este trabajo se asume que los servicios *cloud* soportan comunicaciones seguras sobre HTTPS y permiten la gestión de autorización delegada mediante el protocolo estándar OAuth 2.0. También se presupone que estos servicios se ejecutan en servidores sin limitaciones en término de capacidades de cómputo, memoria, almacenamiento, ancho de banda o consumo de energía, y que se encuentran aislados del resto de otros servidores desplegados en el mismo *cloud*, el cual está ubicado en grandes centros de procesamiento de datos de cualquier parte del mundo.

- Dispositivos *edge* o dispositivos de borde: estos dispositivos pueden ser tanto concentradores, *gateways*, controladores embebidos o independientes, o incluso, conjuntos de dispositivos *edge* trabajando de manera colaborativa o centros de procesamientos de datos pequeños. Las características más significativas que representan este tipo de dispositivos son:

  1. Están ubicados más cerca de los dispositivos IoT que los propios servicios *cloud*.
  2. Esta proximidad les permiten tener una mejor eficiencia y comunicaciones de baja latencia con los dispositivos IoT.

3. Son capaces de almacenar de manera confiable información sensible como contraseñas o *tokens*.
4. Actúan como intermediarios o *proxies* de los dispositivos IoT desplegados en el mundo físico que necesitan interactuar con los entornos *cloud*.

En este caso, se asume que estos dispositivos *edge* soportan también comunicaciones seguras sobre HTTPS para comunicarse con los servicios *cloud*, y por otro lado, también soportan CoAP sobre DTLS [137] para comunicarse de manera segura con los dispositivos IoT. Además, deben ser capaces de soportar credenciales de OAuth 2.0 y de negociar con el servidor de autorización de OAuth 2.0 tanto el *token* de acceso como el *token* de refresco con diferentes roles o *scopes* cuando sea necesario.

- Dispositivos IoT: todo aquel dispositivo embebido dentro del mundo físico y con el mínimo conjunto de recursos y capacidades disponibles para cubrir su función dentro del dominio de aplicación en el que está desplegado estaría incluido dentro de esta categoría. Estos dispositivos no soportan OAuth 2.0, pero pueden ser clientes de una arquitectura que lo soporte a través de la delegación de autorización de los dispositivos *edge* que se tratará a lo largo de este capítulo. Se asume que, dadas sus limitaciones inherentes, estos dispositivos no son capaces de custodiar de manera segura en reposo ningún tipo de secreto o credencial y que no cuentan con la capacidad de ejecutar funciones criptográficas complejas pesadas que provoque la saturación de sus propios recursos o un considerable e inaceptable consumo de su batería. Finalmente, como ya se ha comentado en los dispositivos *edge*, se asume que estos dispositivos tienen la capacidad de comunicarse de manera segura con los dispositivos de borde usando CoAP sobre DTLS [137].

Dada esta arquitectura de referencia y las asunciones tomadas en este trabajo, la delegación de autorización de los dispositivos *edge* propuesta en la siguientes secciones muestra cómo los dispositivos IoT pueden interactuar con los servicios *cloud* a través de los dispositivos de borde que soportan OAuth 2.0 y cómo estos dispositivos pueden delegar su autorización en dichos dispositivos IoT. Además,

dentro de este trabajo también se cubre cómo los servicios *cloud* pueden enviar acciones u órdenes que requieren que realicen los dispositivos IoT y cómo este flujo de actuación se apoya en un esquema de nombres jerárquico para que los propios servicios *cloud* tengan un mejor entendimiento de qué dispositivo o conjunto de ellos están ubicados en qué localización sobre la que se necesita cambiar el estado del mundo físico que las rodea.

3.3. Tecnologías y especificaciones fundamentales

3.3.1. OAuth

OAuth ha sido desarrollado con el fin de permitir el control de acceso en sistemas heterogéneos a recursos protegidos. Este control de acceso no sólo cubre el del propio propietario del recurso en sí, sino también, permite la delegación de autorización en el acceso a los recursos protegidos por parte de terceros previo consentimiento del propietario de dichos recursos.

Las dos características más importantes que introduce OAuth como protocolo de autorización delegada en sistemas distribuidos son las siguientes:

- Realiza una separación clara de roles como son el propietario del recurso y el cliente (o tercero) que quiere obtener acceso a los recursos protegidos en nombre del propietario de dichos recursos.

- Introduce una capa independiente de autorización gestionada de manera centralizada en los recursos HTTP. Esta independencia de la capa de autorización, desde un punto de vista muy simplista, se asemeja en cierto modo a la definida en el estándar XACML [8] (*eXtensible Access Control Markup Language*) englobado dentro de la gestión de control de acceso basado en atributos ABAC [7] (*Attribute-Based Access Control*).

Otros protocolos de federación de identidades ampliamente utilizados a día de hoy como SAML [138] y OpenID Connect [139] ponen foco principalmente en la autenticación denominada *Single Sign-On* [140] (SSO), es decir, autenticar al usuario una única vez para todas sus aplicaciones permitidas eliminando así,

la necesidad de volver a introducir las credenciales cuando el usuario cambia de aplicación durante una sesión particular. Sin embargo, desde el punto de vista de autorización en el acceso, dicha gestión se basa en la identidad del usuario autenticado, es decir, es el mismo usuario, aplicación o servicio autenticado el que con su propia identidad accede a los recursos protegidos para los que está autorizado.

Sin embargo, la característica principal de autorización delegada de OAuth permite que una aplicación tercera pueda acceder a los recursos protegidos de un usuario en su nombre mediando previamente el consentimiento de dicho usuario, lo cual habilita una gestión de control de acceso ya no sólo del usuario autenticado como con los protocolos anteriores, sino de un tercero cuyo acceso está acotado al consentimiento que le ha dado el propietario de los recursos. Aunque esto es algo que se lleva extendiendo años gracias a Google, Facebook, Twitter o Github, a día de hoy, y debido a la directiva Europea de PSD2 [141] (*Payment Services Directive* (EU) 2015/2366), dicho control de acceso basado en la autorización delegada trasciende incluso a los entornos financieros, los cuales se apoyan en implementaciones estándares a nivel industria basadas en OAuth, como las definidas por el *Berlin Group* [142], con el fin de cubrir dicha directiva Europea.

Especificaciones y arquitectura de los componentes de OAuth 2.0

El protocolo OAuth vió la luz por primera vez con su especificación inicial 1.0 [143] en diciembre de 2007. Casi dos años después, en junio de 2009, se creó la especificación 1.0a [144] para solventar una vulnerabilidad [145] existente en la primera versión del protocolo relacionada con un ataque de tipo fijación de sesión contra la solicitud de *token* dentro del flujo de aprobación. Esta versión 1.0a se apoyaba fuertemente en la criptografía, especialmente en firmas digitales, lo que le permitía no depender de la capa de transporte HTTPS/TLS para evitar ataques MitM (*Man in the Middle*) ya que como cada mensaje era firmado individualmente, cualquier construcción o firma incorrecta de un mensaje provocaba la invalidación de la transacción completa. Esto, aunque seguro, se convertía en un reto para la mayoría de desarrolladores a la hora de implementar dicha especificación.

En octubre de 2012, se liberó la versión 2.0 de OAuth [3] que es la más exten-

dida en el mercado a día de hoy. En comparación con su versión anterior, OAuth 2.0 desecha la gran carga criptográfica que tenía su versión predecesora convirtiendo esta nueva versión en una especificación centrada en *tokens* al portador (en inglés, *bearer tokens*) que no proporcionaban mecanismos internos de seguridad y que podían ser copiados o robados al depender OAuth 2.0 de la capa de transporte HTTPS/TLS para evitar ataques MitM. Estas características, aunque repercuten negativamente en la robustez del protocolo desde el punto de vista de seguridad y delegan en la implementación de la especificación las características de seguridad, permiten que dicha especificación sea más fácil de implementar y por este motivo, se ha acelerado tanto su adopción por parte de la industria. Otra característica novedosa de esta nueva versión es la flexibilidad que permite el protocolo, ya que en esta versión no sólo se permite la gestión de flujos con aplicaciones web como su predecesora sino que también incluye clientes no web. Además, como ya se ha comentado con anterioridad en el desarrollo de este capítulo, se incluye una mejor separación de roles para llevar a cabo una correcta gestión de la autorización y consentimiento relacionado con el control de acceso delegado.

Dentro de la separación de roles especificada en OAuth 2.0 se definen los siguientes cuatro:

- Servidor de Autorización (en inglés, *Authorization Server*): este rol define el servidor central de todo el esquema de OAuth 2.0 mediante el cual se materializa la gestión de consentimiento y la delegación de autorización entre un cliente y el propietario de los recursos. Los consentimientos que otorgan los propietarios de los recursos a través del servidor de autorización se desglosan a un nivel de granularidad que permite especificar las capacidades de acceso que dispondrá el cliente sobre el servidor de recursos. Una vez aprobado dicho consentimiento por parte del propietario de los recursos, el servidor de autorización emite un *token* de acceso (en inglés, *access token*) con un periodo de vida limitado al cliente mediante el cual, dicho cliente podrá accedera los recursos protegidos custodiados en el servidor de recursos en nombre del propietario de los mismos. Además, y según el flujo de OAuth 2.0 que se utilice, el servidor de autorización podría emitir junto al *token* de acceso, un *token* de refresco (en inglés, *refresh token*), el

Tabla 3.1: Roles existentes en la arquitectura *edge-centric* de referencia

Rol OAuth	**Rol en la solución propuesta**
Propietario del recurso + Servidor de recursos	Servicio *cloud*
Cliente	Dispositivo IoT (a través del dispositivo *edge*)
Servidor de autorización	Servidor OAuth

cual permitiría al cliente volver a solicitar un *token* de acceso nuevo cuando el anterior caducará sin necesidad de volver a iniciar el flujo de gestión de consentimiento con el propietario del recurso.

- Servidor de Recursos (en inglés, *Resource Server*): es el servidor que se encarga de custodiar los recursos protegidos a los que, previo consentimiento de los propietarios de los recursos, el cliente intenta acceder mediante los *tokens* de acceso. Obviamente, no sólo los clientes pueden acceder a dichos recursos protegidos, ya que gracias a los distintos flujos definidos en OAuth 2.0, los propietarios por sí mismos también podrían acceder a dichos recursos del mismo modo que los clientes, es decir, mediante *tokens* de acceso.

- Cliente (en inglés, *Client*): es la aplicación de terceros (aplicación web tipo javascript, aplicación móvil nativa o híbrida, servidor, etc.) que accede a los recursos protegidos en nombre del propietario de dichos recursos previo consentimiento de éste. Sin la autorización expresa materializada vía consentimiento de a qué puede o no puede acceder un cliente, un cliente por sí mismo no podría acceder a ningún recurso protegido almacenado y custodiado por el servidor de recursos.

- Propietario del recurso (en inglés, *Resource Owner*): dentro de la propia delegación de autorización al rol de cliente descrita en el punto anterior, el rol de propietario del recurso sería el rol más importante ya que es el encargado de otorgar su consentimiento con el objetivo de conceder acceso a un subconjunto (o a la totalidad según el caso) de sus propios recursos protegidos.

La tabla 3.1 muestra la correspondencia entre los roles de la especificación tradicional de OAuth 2.0 y los roles considerados dentro de la arquitectura de referencia planteada en este trabajo.

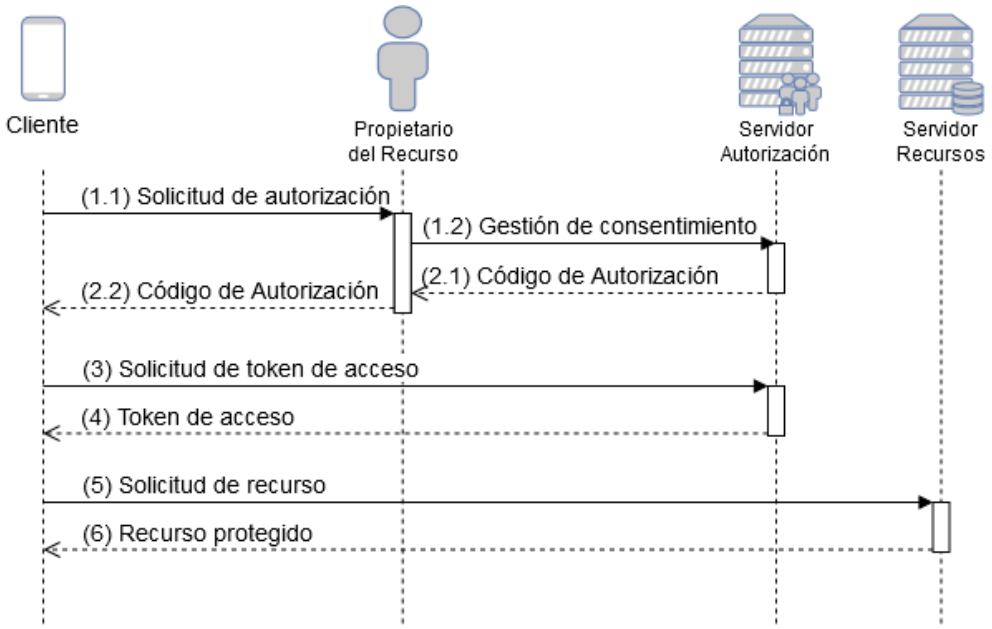

Figura 3.2: Flujo de autorización a alto nivel de OAuth 2.0

Volviendo a la especificación formal estándar de OAuth 2.0, para que un cliente pueda ganar acceso a los recursos protegidos gracias al consentimiento otorgado por el propietario de dichos recursos, la especificación de OAuth 2.0 define cuatro flujos de autorización. El flujo de autorización a alto nivel de OAuth 2.0 mostrado en la figura 3.2 consta de los siguientes pasos:

1. El cliente, al no disponer de autorización previa, solicita al propietario de los recursos dicha autorización. Es este propietario de los recursos protegidos quien gestiona su consentimiento con el servidor de autorización para otorgar acceso al cliente.

2. Una vez aprobado dicho consentimiento, el servidor de autorización emite un código de autorización que refleja dicho consentimiento y se lo retorna al propietario de los recursos protegidos. Finalmente, dicho código de autorización es remitido al cliente.

3. El cliente, una vez que dispone del código de autorización, se identifica en el servidor de autorización para solicitar el intercambio de dicho código por

el *token* de acceso.

4. Validado el cliente y el código de autorización presentado por éste, el servidor de autorización emite un *token* de acceso con un tiempo de vida limitado y se lo retorna al cliente.

5. Haciendo uso del *token* de acceso obtenido, el cliente solicita al servidor de recursos el recurso protegido.

6. Finalmente, validado el *token* de acceso en el servidor de recursos, se concede acceso al recurso protegido al cliente, el cual podrá acceder a este recurso protegido u otros siempre que esté autorizado para ello y su *token* siga siendo válido.

Dependiendo del tipo de la aplicación cliente, este flujo a alto nivel descrito para la obtención de la autorización delegada puede materializarse de diferentes formas dentro de la especificación de OAuth 2.0. Dentro de dicha especificación del protocolo, se identifican cuatro escenarios distintos que cubren este aspecto:

- Flujo *Authorization Code*: este es el flujo que mejor representa el protocolo de OAuth 2.0 ya que en él se involucran todos los roles principales y se realiza una gestión de autorización y consentimiento robusta. En primer lugar, el cliente solicita autorización para acceder a los recursos protegidos al propietario de los mismos, quién a través del servidor de autorización, se identifica como propietario de los recursos solicitados y gestiona su consentimiento para otorgar acceso a dicho cliente en concreto. Este consentimiento se materializa en forma de un código de autorización que le es devuelto finalmente al cliente. Una vez que dicho cliente está en disposición del código de autorización, éste se identifica como el cliente concreto al que le concedieron el consentimiento y presenta el código de autorización en su solicitud de nuevo al propio servidor de autorización para que le sea emitido el *token* de acceso asociado. Si todo este flujo se lleva a cabo de manera satisfactoria, el cliente podrá acceder a los recursos protegidos en nombre del propietario de los recursos con dicho *token* de acceso obtenido.

- Flujo *Implicit*: en el caso de que la aplicación cliente sea una aplicación web tipo javascript se utilizará este flujo de autorización de OAuth 2.0. Dicho flujo es muy similar al flujo *Authorization Code* pero con una salvedad debido a la naturaleza intrínseca de la aplicación cliente: la aplicación, al ser una web, no puede custodiar credenciales de manera segura ya que todo su código fuente se encuentra en el navegador de quien la descargue. Bajo esta premisa, cuando dicha aplicación solicita autorización de acceso al cliente y éste gestiona su consentimiento en el servidor de autorización, se emite directamente el *token* de acceso en vez del código de autorización, para que la aplicación cliente pueda comenzar a consumir recursos protegidos con él sin necesidad de identificarse como aplicación cliente.

- Flujo *Resource Owner Password Credentials*: cuando existe un alto grado de confianza entre el cliente y el propietario del recurso, éste último le cede sus credenciales al cliente para que se identifique en el servidor de autorización como cliente válido adjuntando también en dicha solicitud, las credenciales del propietario de los recursos. Por este motivo, no existe ningún tipo de gestión de consentimiento por parte del propietario del recurso previa a la obtención del *token* de acceso por parte del cliente ya que al ceder éste sus credenciales, no se le involucra dentro del flujo de obtención del *token* de acceso.

- Flujo *Client Credentials*: este flujo se utiliza cuando el propietario del recurso quiere acceder a sus propios recursos protegidos. En lugar de utilizar un modelo de autenticación distinta, el propietario del recurso se autentica en el servidor de autorización y obtiene un *token* de acceso que le permite interactuar con el servidor de recursos de la misma forma que cualquier cliente.

Consideraciones de seguridad de OAuth 2.0

La facilidad de implementación y flexibilidad de OAuth 2.0 al desechar las características criptográficas más robustas a la par que complejas en comparación con su versión 1.0a, provoca que la seguridad sea delegada en la propia implemen-

tación que cada desarrollador realice a partir de la especificación. Es por eso que la definición del propio protocolo OAuth 2.0 [3] cuenta con un amplio conjunto de consideraciones de seguridad dentro de su definición. Dichas consideraciones se centran principalmente en limitar los privilegios de acceso que otorga el *token* de acceso, limitar el tiempo de vida de los distintos *tokens* y códigos incluidos dentro de la especificación, y remarcar la importancia de la custodia segura tanto en tránsito como en reposo de dichos *tokens* y de las diferentes credenciales utilizadas para autenticarse tanto por el cliente como por los distintos propietarios de los recursos.

Con el fin de poner aún más énfasis en la importancia de un desarrollo seguro a la hora de implementar un servidor de autorización que soporte OAuth 2.0, en enero de 2013 vió la luz una especificación focalizada en el modelo de amenazas y consideraciones de seguridad de OAuth 2.0 [146]. En primer lugar, si se pone foco en el modelo de amenazas descrito en dicha especificación, aunque se realiza con un nivel granularidad que llega a rol y flujo de autorización de OAuth, se puede observar una preocupación que aúna todas las amenazas bajo una sola: la fuga de información sensible, entendiéndose como información sensible los distintos tipos de *token*, códigos de autorización y credenciales incluidos dentro de la especificación de OAuth 2.0. Esto materializa la preocupación existente por parte de la comunidad de autores de la especificación de OAuth 2.0 al perder las características criptográficas de los *tokens* de su versión predecesora que permitían una mayor robustez del protocolo con independencia de la seguridad que aportase la capa de transporte gracias a HTTPS/TLS. Por otro lado, con respecto a las consideraciones de seguridad descritas a continuación del modelo de amenazas en la especificación, además de reincidir en la necesidad de una custodia segura tanto en tránsito como en reposo de los distintos *tokens* y credenciales apoyándose en capacidades criptográficas, se esboza una consideración de seguridad bastante interesante extraída de la especificación del protocolo SAML: vincular los *token* al portador a una audiencia específica, es decir, a un cliente concreto. Con esta consideración se busca minimizar el impacto en caso de robo o pérdida de dicho *token* ya que éste sólo podría ser válido al ser utilizado por ese cliente concreto al que se vinculó. Es esta recomendación la que rompe por fin dentro de OAuth con la necesidad inevitable del uso de criptografía como única alternativa para

contextualizar o asegurar que un cliente que utiliza un *token* es quién dice ser y sólo él lo usa de manera legítima.

El nacimiento del protocolo OpenID Connect [139] en 2014, el cual usa como base los flujos y características de OAuth 2.0, no hizo más que reforzar la necesidad de la contextualización de los distintos *tokens* al portador de OAuth de la misma forma que se especifica para el *token* de identidad de OpenID Connect. Es por ello, que apenas un año después, en 2015 se publicó la especificación de introspección de *token* de OAuth 2.0 [147], la cual buscaba principalmente obtener a partir de un *token* de acceso los metadatos asociados, siendo éstos en su mayoría, los mismos que se incluyen en el *token* de identidad de OpenID Connect. A día de hoy, distintas implementaciones a nivel industria utilizan el estándar de JWT [148] (*JSON Web Token*) para los *tokens* de acceso de OAuth 2.0 con el fin de incluir en dichos *tokens* los metadatos asociados del mismo modo que OpenID Connect en su *token* de identidad. Esto permite robustecer los *tokens* de OAuth centrándose en el contexto alrededor del *token* y delegando el uso de las capacidades criptográficas para la firma de dichos JWT en el servidor de autorización, el cual hace uso de ellas exclusivamente durante la emisión y posterior validación de los distintos *tokens*.

OAuth en el Internet de las Cosas

En la recopilación de trabajos incluidos en la tabla 2.2 se puede observar que la mayor parte de los mecanismos que implementan el control de acceso basado en *tokens* se apoyan en el protocolo OAuth 2.0 para resolver la autorización basada en el contexto alrededor del dispositivo. El uso de OAuth 2.0 como protocolo de autorización en dispositivos IoT obviamente no se reduce a los trabajos recopilados en dicha tabla sino que podemos encontrarlo en infinidad de ejemplos más [149–155]. Esto es así ya que, como ya se ha comentado, OAuth 2.0 es un protocolo que permite una alta flexibilidad y simplicidad en el control de acceso llevando a cabo dicho control en tiempo real. Sin embargo, el uso de OAuth 2.0 en el IoT también supone dos grandes retos a la hora de ser implantado dada sus características intrínsecas. Por un lado, la gestión de autenticación de los distintos dispositivos se basa, como indica en su especificación, en credenciales y *tokens*.

Dentro de las consideraciones de seguridad ya comentadas, se observaba la imperiosa necesidad por parte del equipo de definición de la especificación de OAuth de una custodia segura tanto en tránsito como en reposo de las credenciales y *tokens*, lo que, dadas las características de los dispositivos y sus limitaciones de recursos supone un reto de base. Por otro lado, el protocolo OAuth 2.0 define en sus flujos principales una gestión de consentimiento para otorgar la delegación de autorización al dispositivo que accederá en su nombre a sus recursos protegidos. Esta intervención manual por parte de un usuario en un entorno con una cantidad ingente de dispositivos y que pueden no tener la interfaz suficiente para dicha interacción humana, convierte este caso en el segundo gran reto del uso de OAuth 2.0 en el mundo del Internet de las cosas.

En función del caso de uso, el almacenamiento seguro de credenciales en los dispositivos se intenta abordar de múltiples formas. Una de las más representativas y más ampliamente extendida dentro del Internet de las cosas es el uso de funciones PUF (*Physically Unclonable Functions*). Estas funciones se basan en las características intrínsecas de los circuitos hardware del dispositivo concreto del cual forman parte y le proporcionan un mecanismo criptográfico para identificarse de manera unívoca [156, 157]. Dichas funciones podrían describirse de manera más sencilla como la huella dactilar del hardware, la cual no se puede clonar y en el caso de intentar manipularla, dicha huella se corrompe dejando de tener validez. Otras opciones utilizadas para realizar el almacenamiento seguro de credenciales en dispositivos, aunque estas ya requieren de que el dispositivo de recursos limitado tenga de base algo más de capacidad que los que se apoyan en las funciones PUF anteriormente citadas, se basan en características específicas proporcionadas por los sistemas operativos hechos a medida para los dispositivos del Internet de las cosas como por ejemplo CerberOS [158] o incluso, si el dispositivo tiene todavía algo más de capacidad, se apoyan en claves criptográficas y certificados digitales [159, 160].

Finalmente, y referido a la gestión de consentimiento por parte del propietario del recurso dentro de los flujos de delegación de autorización de OAuth 2.0, existen diversas iniciativas que intentan extender dicho protocolo para permitir una gestión de consentimiento dinámica como son la iniciativa del IETF OAuth 2.0 *Device Flow* [161] y la iniciativa del grupo Kantara de *User-Managed Ac-*

cess [162]. Ambas iniciativas intentan definir flujos de OAuth 2.0 mediante los cuales la interacción del propietario del recurso para otorgar su consentimiento sea la menor posible sin llegar a eliminarse por completo, es decir, no buscan la automatización completa de la gestión de consentimiento del usuario. Por este motivo, ambas iniciativas, aunque ambiciosas, siguen requiriendo de dicha interacción humana en sus flujos lo que provoca que el factor limitante de la existencia de procesos manuales impida una automatización completa en los entornos altamente escalables del Internet de las cosas.

A modo resumen de todo lo ya descrito a la hora de aplicar OAuth en el ámbito del Internet de las cosas cabe destacar que; la propia flexibilidad que permite dicho protocolo de ser extendido junto con la viabilidad demostrada por los diferentes trabajos anteriores de poder utilizarlo para abordar los diferentes retos que plantea su implantación en el mundo heterogéneo y altamente escalable del Internet de las cosas ha propiciado que OAuth 2.0 sea la opción elegida en esta tesis para cubrir las distintas necesidades de gestión de control de acceso a recursos *cloud* por parte de dispositivos con recursos limitados a través de los denominados dispositivos *edge*.

3.3.2. CoAP

Una vez seleccionado OAuth 2.0 como el protocolo que se usará en este trabajo para llevar a cabo el control de acceso basado en *tokens* e identificados sus puntos clave con el fin de adecuarlo a las necesidades del Internet de las cosas, se hace necesario identificar el protocolo a nivel de aplicación que mejor encaja con las características intrínsecas y limitaciones de los dispositivos IoT. Partiendo de la base que OAuth 2.0 requiere de HTTP para poder ser implantado, CoAP representa una estructura similar a la de HTTP que ayuda perfectamente a encajar el mundo del Internet de las cosas a los entornos *cloud* a través de los dispositivos *edge*.

CoAP [38] es un protocolo que a diferencia de HTTP, utiliza UDP como protocolo de transporte en vez de TCP. Esto le confiere una serie de características muy beneficiosas que encajan perfectamente con dispositivos de recursos limitados. Entre dichas características, el mero hecho de utilizar UDP le proporciona a

CoAP la independencia de mantener comunicaciones síncronas, lo que le permite, entre otras características, hacer un uso más eficiente de recursos tan importantes como es la batería para los dispositivos IoT. Además, esta particularidad también le permite poder utilizarse no solo en arquitecturas cliente-servidor sino también en modelos de publicación-suscripción soportando envío de mensajes multicast. Por otro lado, el consumo de recursos que realiza el propio protocolo en la comunicación para llevar a cabo el correcto funcionamiento (también denominado *overhead*) es mucho más simple y ligero que HTTP, lo cual lo convierte en la solución idónea desde el punto de vista de eficiencia en el uso de recursos [163].

Finalmente, el uso de CoAP por parte de los dispositivos IoT permite a los dispositivos *edge* realizar una traducción de manera sencilla de dicho protocolo a HTTP para comunicarse con el *cloud*. Esto es así debido a que CoAP se estructura de una manera similar a HTTP, es decir, se utilizan URIs (*Uniform Resource Identifier*) para identificar los diferentes recursos dentro del sistema y los mismos verbos que HTTP como métodos para el acceso a los diferentes recursos, lo cual permite facilitar el uso de OAuth 2.0 en este tipo de arquitecturas.

3.4. Solución propuesta para la gestión de identidades y accesos en IoT

Una vez expuestas las tecnologías y especificaciones fundamentales que sirven de eje principal del trabajo desarrollado a lo largo de esta tesis, se describe en esta sección la solución diseñada para la gestion de identidades y accesos de los dispositivos IoT. Dicha solución define tres flujos principales: un flujo de registro en el sistema y dos flujos o modalidades de acceso a los recursos protegidos ubicados en el *cloud* por parte de los dispositivos IoT.

3.4.1. Registro de dispositivos IoT

El flujo de registro de dispositivos IoT es el primer flujo planteado dentro de este trabajo. Gracias a él, se da soporte tanto a los flujos de control de acceso a recursos ubicados en el *cloud* por parte de dichos dispositivos IoT como a la propia

gestión de direccionamiento del flujo de actuación para poder enviar órdenes o solicitudes en la dirección inversa, es decir, del *cloud* a los dispositivos.

Flujo de registro

El flujo de registro de un dispositivo IoT en un sistema debe cubrir las dos características principales que definen este tipo de ecosistemas: la limitación inherente de recursos y capacidades de dichos dispositivos, y la necesidad de gestión de la alta escalabilidad a la que se enfrenta un sistema que intenta abordar un dominio de aplicación apoyado en esta clase de tecnología.

En primer lugar, este flujo de registro evita la necesidad de llevar a cabo un proceso de autenticación robusta por parte de un dispositivo IoT en el sistema ya que esto provocaría un mayor consumo de recursos y podría darse el caso que dicho dispositivo no dispusiera de las capacidades necesarias para llevar a cabo dicho proceso de autenticación robusta, lo que le dejaría de inicio fuera del sistema. Con el fin de cubrir el mayor espectro de dispositivos identificando y autenticando cada uno de ellos aún con sus limitaciones de recursos inherentes, este proceso se basa para realizar dichas tareas en la validación de la propia huella del dispositivo, es decir, se apoya en el contexto del dispositivo, incluyendo tanto atributos estáticos como dinámicos [164, 165] para llevar a cabo la emisión de un *token* de identidad que lo identifique. Esta huella del dispositivo puede construirse a partir de las propiedades del hardware del propio dispositivo, dirección física o lógica que dispone al conectarse a la red, su user agent, su geolocalización, o incluso, otras características de su comportamiento identificable al analizar su tráfico de red como los protocolos de red que utiliza, tamaño de cabeceras y mensajes, latencias y tiempos entre mensajes, u otros. Todas estas características del contexto del dispositivo son almacenadas en el *cloud* durante este proceso de registro y se asocian a un identificador opaco y unívoco que se denomina *token* de identidad o id *token*.

El evitar modelos de autenticación robustos permite que un dispositivo *edge*, aún teniendo también limitación de recursos y capacidades pero mucho menores que las de un dispositivo IoT, pueda gestionar el registro de una gran cantidad de dispositivos cubriendo así, la característica de escalabilidad de este tipo de siste-

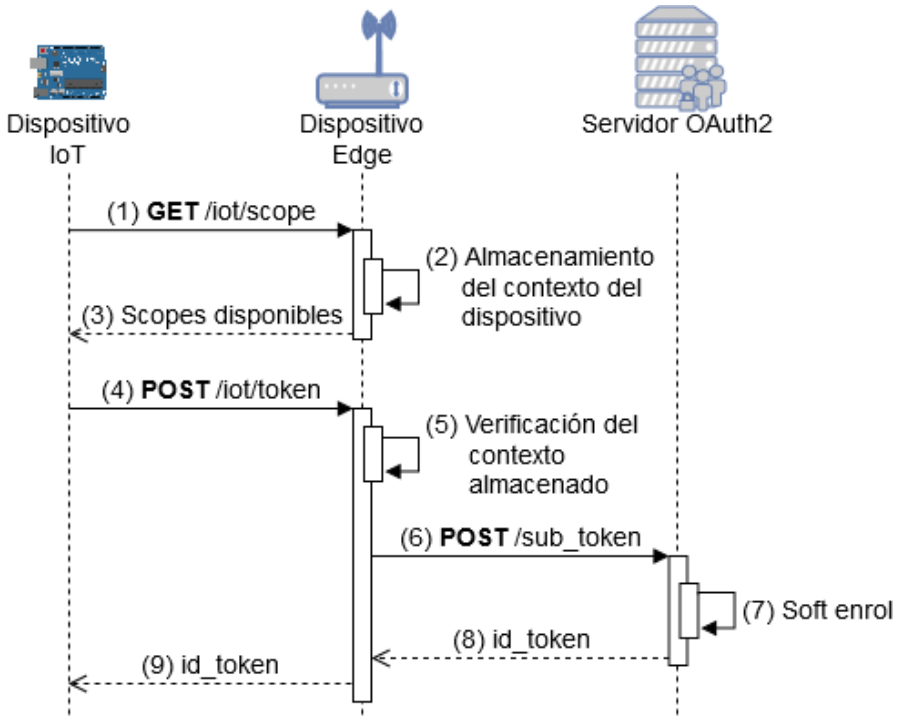

Figura 3.3: Diagrama de secuencia del flujo de registro

mas. Además, el propio proceso de autenticación basado en atributos de contexto del dispositivo y el *token* de identidad posteriormente emitido permiten independizar dicho proceso de los protocolos de comunicación de red subyacentes al no vincularse a ninguna característica concreta de estos protocolos.

En la figura 3.3 se muestra el diagrama de secuencia que cubre este flujo de registro de un dispositivo IoT en un sistema apoyándose para ello en dispositivos *edge*.

1. El dispositivo IoT solicita al dispositivo *edge* el conjunto de roles (o *scopes*) de los que dispone como cliente OAuth 2.0 del sistema. Esta petición es realizada vía CoAP dadas las características del dispositivo IoT y pudiendo ser materializada mediante el método GET o POST en función de la implementación que se realice.

2. El dispositivo *edge* registra y almacena todos los atributos del contexto del dispositivo a partir de la solicitud recibida. Si la solicitud fue realizada me-

diante el método GET de CoAP, el contexto incluiría atributos como la dirección lógica del dispositivo y su user agent en función de las cabeceras utilizadas en la petición. Si por el contrario, la petición fue realizada mediante el método POST de CoAP, dentro del cuerpo del mensaje se podría incluir una mayor cantidad de atributos como la dirección física, la geolocalización u otros.

3. El dispositivo *edge* retorna al dispositivo IoT el conjunto de roles soportados por dicho dispositivo de borde.

4. Una vez el dispositivo IoT ha elegido el conjunto de roles que más se adecuan a su función dentro de los disponibles por el dispositivo *edge* a través del cual se va a registrar en el sistema, solicita su registro en el sistema con el rol específico que necesita para que se le emita un *token* de identidad. Como se puede observar, por diseño, un dispositivo IoT nunca podrá disponer de ningún rol o un mayor privilegio del que disponga el propio dispositivo *edge* a través del cual se va a registrar en el sistema.

5. El dispositivo *edge* valida que el rol solicitado se encuentra entre los que él dispone y verifica que el contexto del dispositivo asociado a la solicitud de *token* de identidad recibida coincide con la que él mismo registró previamente en la solicitud descrita en el punto 1. De no coincidir o no existir registro alguno ya que se ha solicitado el *token* de identidad sin consultar previamente los roles disponibles, la solicitud se descarta.

6. Si el rol es correcto y el contexto se verifica de manera satisfactoria, el dispositivo *edge* solicita al servidor de OAuth 2.0 (el servidor de autorización) la delegación de autorización basada en su propio *token* de acceso. En este momento, el servidor de autorización de OAuth 2.0 vincula la identidad del dispositivo IoT basada en el contexto de dicho dispositivo con el *token* de acceso del dispositivo *edge* para generar un nuevo *token* de identidad que representa la nueva autorización concedida a dicho dispositivo IoT que ha solicitado registrarse en el sistema. Esta delegación puede ser llevada a cabo tantas veces como se necesite por todos los dispositivos que inicien el flujo de registro a través del mismo dispositivo *edge*. Dicha delegación de

autorización jerárquica permite a los servicios *cloud*, más concretamente al servidor de autorización de OAuth 2.0, disponer de la capacidad de revocar *tokens* de identidad puntuales o incluso, todos los *tokens* de identidad asociados a un *token* de acceso concreto simplemente revocando dicho *token* de acceso. Un funcionamiento jerárquico similar de revocación podemos encontrarlo en las infraestructuras de clave pública [166, 167] (PKI - *Public Key Infraestructure*), donde en el caso de revocar una autoridad de certificación (CA - *Certificate Authority*), todos los certificados emitidos por ella quedan automáticamente revocados. En este paso, se asume que el *token* de acceso del dispositivo edge ha sido previamente negociado con el servidor de OAuth 2.0 de manera confiable siguiendo la especificación estándar.

7. El servidor de OAuth 2.0 realiza el registro del dispositivo IoT basado en el contexto que le ha propagado el dispositivo *edge* y vincula dicho contexto al *token* de acceso de este último. Tras esta vinculación, se emite un *token* de identidad de un corto periodo de vida, el cual puede variar dependiendo del dominio de aplicación y caso de uso en el que se despliegue el dispositivo IoT pero nunca debería exceder de una hora o en su defecto, del tiempo de vida máximo del *token* de acceso al que se ha vinculado. Obviamente, dicho *token* de identidad emitido para el dispositivo IoT también queda vinculado con el *token* de acceso del dispositivo *edge* del mismo modo que el contexto del dispositivo anteriormente mencionado.

8. Generado el *token* de identidad, este es devuelto al dispositivo *edge*. Como en cualquier otro modelo basado en *tokens*, el *token* de identidad debe transmitirse siempre a través de canales seguros ya que dicho *token* tiene los permisos y capacidades necesarias para acceder a los recursos protegidos.

9. Finalmente, el *token* de identidad es remitido por canal seguro al dispositivo IoT y por tanto, el procedimiento de registro se da por concluido.

Cabe hacer mención dentro de este flujo de registro de una característica muy importante: la inexistencia de interacción manual por parte de los usuarios para gestionar el consentimiento de OAuth 2.0. Esto es así ya que no se realiza una ampliación del alcance de recursos protegidos a los que se puede acceder por

parte de los dispositivos, sino que es el dispositivo edge quien delega su propia autorización a los dispositivos IoT para que éstos puedan acceder con uno o varios de sus roles permitidos al entorno *cloud*. Este punto, ligado al comentado de los beneficios de llevar a cabo una autenticación e identificación del dispositivo IoT basada en su contexto, permite que este flujo de registro sea cien por cien escalable sin interacción manual que lo limite o restrinja.

3.4.2. Control de acceso a los recursos

En esta sección se hace una clara distinción entre dos modos de acceso a los diferentes recursos protegidos *cloud* por parte de los dispositivos IoT. En primer lugar, se aborda el flujo estándar de acceso a recursos para su creación, modificación, lectura o eliminación de los mismos. En segunda instancia, dadas las características de conectividad de los dispositivos o incluso que dichos dispositivos pueden estar en movimiento dentro del mundo físico en el que se despliegan, se plantea un flujo de control de acceso denominado *roaming* en el cual se cubre la casuística de que un dispositivo IoT, por razón de su movimiento dentro de su entorno, se conecte al sistema a través de otro dispositivo *edge* distinto al que gestionó su registro.

Flujo de acceso

Este flujo define cómo se realiza el control de acceso a los recursos protegidos en el *cloud* sobre los dispositivos IoT una vez que éstos se han registrado correctamente en el sistema y disponen de su *token* de identidad. La potencia que proporciona el control de acceso basado en un *token* de identidad permite cubrir dos aspectos fundamentales en cualquier proceso de autorización delegada: el tiempo de vida (TTL - *Time To Live*) de la autorización concedida que obviamente va ligado al tiempo de vida del *token*, y la trazabilidad de grano fino que permite realizar dicho *token* de identidad sobre el acceso a los distintos recursos protegidos.

Cabe destacar que el *token* de acceso de OAuth 2.0 del que dispone el dispositivo *edge* para acceder a los recursos protegidos en el *cloud* también cuenta con un tiempo de vida útil hasta que éste expira. Una vez dicho *token* de acceso ha ex-

pirado, éste no será válido para acceder a los recursos protegidos y por tanto, será necesario que dicho *token* sea refrescado por otro válido. Obviamente, el tiempo de vida del *token* de acceso va ligado a la propia gestión de autorización y control de acceso del dispositivo *edge* y no a la delegación de autorización que éste realiza hacia los dispositivos IoT, la cual, puede tener un tiempo de vida incluso más corto. El tiempo de vida de esta delegación de autorización permite que el dispositivo IoT pueda crear, acceder, actualizar o eliminar recursos protegidos en el *cloud* a través del dispositivo *edge* gracias al *token* de identidad que el dispositivo IoT obtuvo en el proceso de registro. Cuando este *token* de identidad expira o se revoca debido por ejemplo, a que el contexto del dispositivo IoT ha cambiado significativamente y puede darse el caso de una suplantación de identidad, dicho dispositivo ya no tendrá acceso a ningún recurso creado hasta ese momento aunque vuelva a realizar nuevamente el flujo de registro y obtenga un nuevo *token* de identidad. El nuevo *token* de identidad se considera como una nueva delegación de autorización y no le permitirá acceder a los recursos que ese mismo dispositivo hubiera creado con su *token* anterior. Esto puede ser visto como una gestión de aislamiento de la información que vincula el *token* de identidad del dispositivo IoT con los recursos que él mismo crea en el *cloud* durante el tiempo de vida de dicho *token*. La limitación de acceso que tiene un dispositivo IoT sólo es de aplicación para él, ya que un dispositivo *edge* puede acceder con su *token* de acceso a la información creada por cualquiera de los *tokens* de identidad a los que delegó su autorización o incluso el propio *cloud* pueda llevar a cabo análisis sobre toda la información en su conjunto almacenada en sus servicios.

Esta característica dual de aislamiento de la información y temporalidad de la misma vinculada con el *token* de identidad proporciona una gran capacidad de trazabilidad ya que un *token* de identidad es unívocamente mapeado con un dispositivo IoT y el contexto que había a su alrededor a la hora de solicitarlo y por tanto, a una delegación de autorización específica por parte de un dispositivo *edge* y su *token* de acceso. Gracias a esto, los servicios *cloud* pueden tener conocimiento de qué está realizando cada dispositivo en cada momento.

La figura 3.4 muestra el conjunto de pasos que se realizan dentro de este flujo de control de acceso de los dispositivos a los recursos protegidos en el *cloud*:

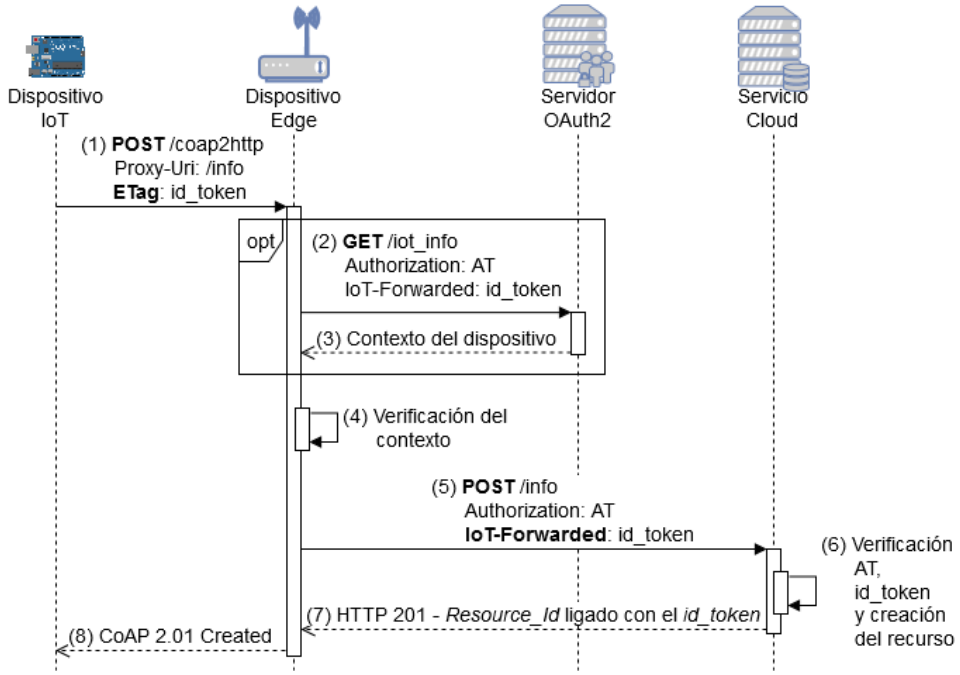

Figura 3.4: Diagrama de secuencia del flujo de acceso

1. El dispositivo IoT realiza a través de su dispositivo *edge* una petición POST de CoAP incluyendo dos cabeceras de dicho protocolo: *Proxy-Uri* y *ETag*. La cabecera *Proxy-Uri* en este caso de uso hace referencia a la URI del servicio *cloud* a donde el dispositivo IoT necesita acceder. Por otro lado, la cabecera *ETag* incluye el *token* de identidad del dispositivo IoT obtenido durante el flujo de registro. Se utiliza esta cabecera para enviar el *token* de identidad ya que, por definición, un *ETag* es un identificador opaco que se asigna a una versión específica de un recurso, y dada la característica ya descrita del aislamiento en el acceso a los datos y su vinculación con dicho *token* de identidad, esta cabecera descrita en la especificación del protocolo CoAP [38] encaja perfectamente con la especificación del *token* de identidad definida en esta tesis. Por este motivo, no es necesario la modificación de la especificación estándar de CoAP [38] para definir y crear una nueva cabecera que cubra las necesidades de este trabajo

2. Como se puede observar en la figura 3.4, este paso es opcional ya que de-

pende del comportamiento de la política de retención y almacenamiento de contextos de los dispositivos IoT que tenga el dispositivo *edge*, es decir, si el contexto está disponible por ejemplo en la caché del dispositivo *edge*, este paso no es necesario. Si por el contrario dicha información no se encuentra disponible, el dispositivo *edge* solicita la recuperación de dicho contexto al servidor de OAuth 2.0 usando el *token* e identidad incluido en la cabecera CoAP *ETag* mencionada en el punto anterior. Sin embargo, en la traducción de CoAP a HTTP, dicho *token* de identidad no se propaga en la cabecera HTTP *ETag* sino en una nueva cabecera denominada: *IoT-Forwarded*. Esta nueva cabecera HTTP permite la coexistencia de la funcionalidad completa de HTTP en el servidor de OAuth 2.0 enriqueciendo la definición estándar de OAuth 2.0 sin convertirla en incompatible para futuras versiones o implementaciones.

3. Si el paso anterior se llegó a ejecutar, gracias al *token* de identidad proporcionado por el dispositivo *edge* al servidor OAuth 2.0, éste es capaz de recuperar el contexto del dispositivo con el cual está vinculado dicho *token* y le devuelve dicha información al dispositivo *edge*.

4. El dispositivo *edge* verifica que el contexto de la petición que ha recibido del dispositivo IoT coincide con el disponibilizado por el propio servidor de OAuth 2.0 en el caso de haber ejecutado los dos pasos anteriores, o con el que tenga almacenado por ejemplo, en su caché.

5. Si la verificación del contexto tiene éxito, el dispositivo *edge* traduce el método de la petición CoAP al correspondiente método HTTP y solicita la URI incluida en la cabecera CoAP *Proxy-Uri* al servicio *cloud*. En este caso, tanto el *token* de acceso del dispositivo *edge* como el *token* de identidad del dispositivo IoT son obligatorios en la petición al servicio *cloud* para poder acceder o crear recursos. En el ejemplo mostrado en la figura 3.4 se puede observar la traducción del método CoAP POST al método HTTP POST.

6. En este punto, primero, el servicio *cloud* verifica los roles o *scopes* del *token* de acceso presentado por el dispositivo *edge* así como su tiempo de

expiración cumpliendo con el estándar de acceso a los recursos de la especificación de OAuth 2.0. Una vez se ha llevado a cabo dicha validación, el siguiente paso es comprobar si el *token* de identidad está vinculado con el *token* de acceso, es decir, si el *token* de identidad fue generado a partir de una delegación de autorización del *token* de acceso presentado. Si estas comprobaciones también son correctas y el método HTTP es un POST, el servicio *cloud* creará el recurso solicitado vinculando el identificador del recurso con el *token* de identidad recibido. En el caso de que el método HTTP sea un GET, un PUT o un DELETE, este paso es sutilmente diferente ya que el servicio *cloud* verificará primero que el *token* de identidad es el mismo que creó el recurso por primera vez y que por tanto, ambos identificadores están vinculados. Si esta verificación es correcta, el dispositivo IoT a través del dispositivo *edge* podrá acceder, modificar o eliminar el recurso de su propiedad.

7. El servicio *cloud* responde al dispositivo *edge* con la respuesta HTTP apropiada dependiendo de los resultados del paso anterior de este flujo. Por ejemplo, en la figura 3.4, el servicio *cloud* produce una respuesta *HTTP 201 Created* ya que la solicitud fue la de creación de un recurso a través del método HTTP POST.

8. Finalmente, el dispositivo *edge* traduce la respuesta HTTP recibida a la correspondiente respuesta CoAP con vista a actuar de intermediario y propagar la información solicitada al dispositivo IoT.

Flujo de acceso en *roaming*

Cuando un dispositivo IoT pierde su conexión con su dispositivo *edge* ya sea, por ejemplo, debido a limitaciones de cobertura, potencia de señal débil o incluso, que el propio dispositivo IoT se encuentra en movimiento dentro del mundo físico en el que es desplegado, éste podría conectarse a otro dispositivo *edge* distinto más cercano con mejor señal para seguir dando servicio. En este punto, el dispositivo IoT ya dispone de un *token* de identidad válido aunque dicho *token* fue generado durante su proceso de registro con otro dispositivo *edge*. Por este motivo, este

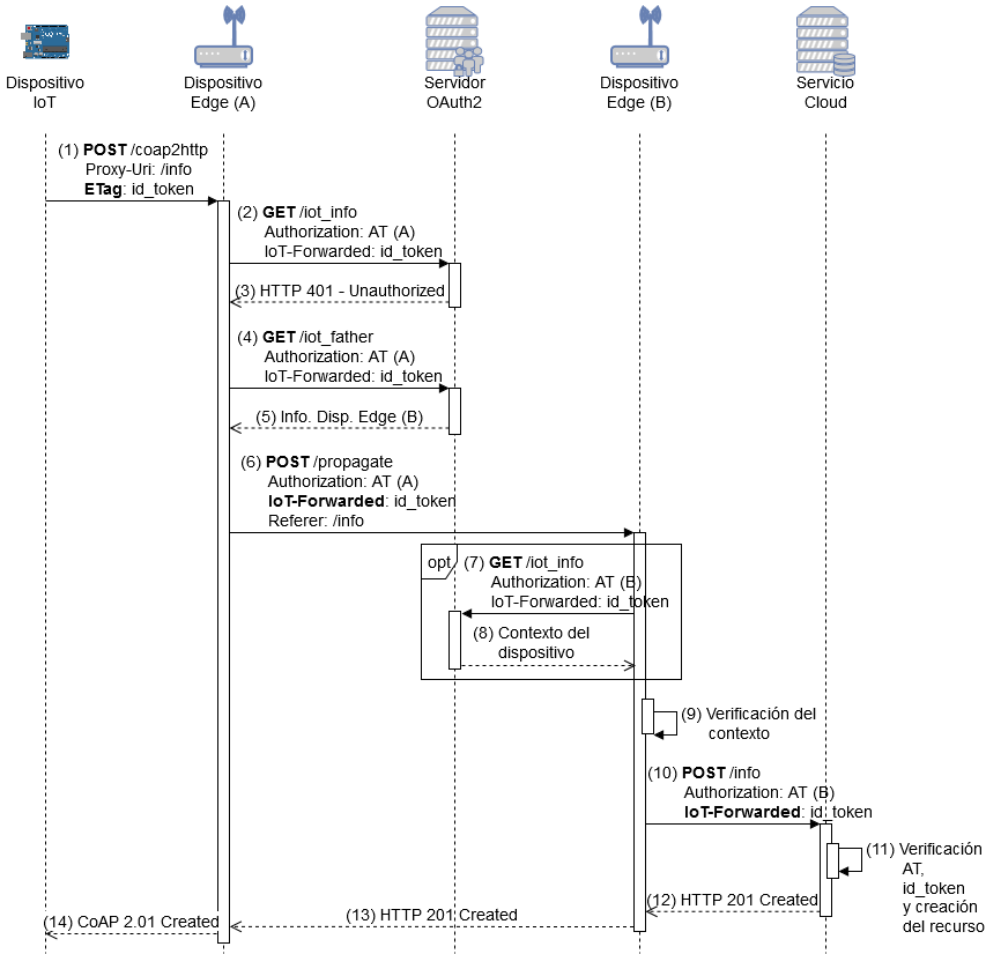

Figura 3.5: Diagrama de secuencia del flujo de acceso en *roaming*

flujo plantea una extensión del flujo de acceso descrito en la sección anterior con el objetivo de abordar esta situación.

En la figura 3.5 se muestra el flujo propuesto para el control de acceso a recursos por parte de dispositivos IoT en *roaming*. Como se puede observar en dicha figura, el nuevo dispositivo *edge* A no dispone de información del contexto del dispositivo IoT cuando éste intenta utilizarlo para consumir recursos en el *cloud*. Por este motivo, el dispositivo *edge* A solicita al servidor de OAuth 2.0 dicha información pero como el *token* de identidad del dispositivo IoT no se generó a partir de una delegación de autorización del *token* de acceso del dispositivo *edge* A, el servidor de OAuth 2.0 no le retornará dicho contexto como se puede apreciar

en los pasos 2 y 3 del flujo.

En este punto, la única opción que le queda al dispositivo *edge* A es averiguar qué otro dispositivo *edge* vinculó su *token* de acceso con el *token* de identidad presentado (paso 4 del flujo). Para este caso, se ha añadido una nueva funcionalidad al servidor de OAuth 2.0 mediante la cual un dispositivo *edge* con un *token* de acceso autorizado puede solicitar la información de la delegación de autorización realizada por otro dispositivo *edge* mediante la consulta de quién es el ”padre” del *token* de identidad incluido en la solicitud.

Con esta información, el dispositivo *edge* A propaga la solicitud del dispositivo IoT usando el método HTTP POST al dispositivo *edge* B (paso 6 del flujo). Dentro del cuerpo del mensaje propagado vía HTTP POST por el dispositivo *edge* A se incluye el contexto actual del dispositivo IoT con el fin de poder realizar las verificaciones necesarias y también, la cabecera HTTP referrer que especifica la URI del servicio *cloud* solicitada por el dispositivo IoT para la cual el dispositivo *edge* A no tuvo permisos suficientes para acceder. A partir de este punto, y hasta el paso 12 del flujo, éste continúa como se ha descrito en el flujo de acceso estándar hasta que finalmente, en el paso 13, el dispositivo *edge* B envía la respuesta del servicio *cloud* al dispositivo *edge* A para garantizar que dicha respuesta llega finalmente al dispositivo IoT, que fue quien inició el flujo de acceso en *roaming* a través de este nuevo dispositivo *edge* A.

Adicionalmente a esta funcionalidad principal de acceso en *roaming*, este flujo permite desplegar arquitecturas de enjambres de dispositivos *edge* que colaboran entre sí para dar una mejor cobertura a los dispositivos IoT desplegados en el mundo físico. Esto puede incrementar en algunos casos la latencia y la sobrecarga de la red pero al mismo tiempo mejora, entre otros:

- El rendimiento del sistema ya que un dispositivo IoT siempre buscará siempre el dispositivo *edge* más cercano que más recursos disponibles tenga.
- La capacidad de balanceo de carga entre diferentes dispositivos *edge* para evitar la alta densidad de las zonas de despliegue.
- La resiliencia del sistema adaptándose mejor a los diferentes cambios de las topologías del mismo.

3.5. Solución propuesta para el direccionamiento y nombrado en IoT

Hasta el momento se ha abordado cómo se lleva a cabo la creación, acceso, modificación y eliminación de recursos protegidos por parte de los dispositivos IoT en los servicios *cloud* a través de los dispositivos *edge*. Sin embargo, resta detallar cómo los diferentes servicios *cloud* realizan la comunicación inversa, es decir, cuando un servicio *cloud* necesita actuar sobre el mundo físico para modificar su estado, cómo éste manda órdenes sobre todos (o una parte de) los dispositivos IoT desplegados en el sistema. Para ello, en esta sección se abordará tanto el flujo de actuación mediante el cual se lleva a cabo este proceso, así como el esquema de nombrado utilizado por el *cloud* para saber sobre qué conjunto de los dispositivos IoT disponibles en el sistema quiere realizar una u otra acción.

3.5.1. Flujo de actuación basado en direccionamiento dirigido por eventos

Este flujo de actuación, del mismo modo que los descritos anteriormente de acceso a los recursos protegidos, requiere que el dispositivo IoT se haya registrado en el sistema a través de un dispositivo edge y disponga por tanto, de su *token* de identidad para interactuar con los servicios *cloud*. El disponer de *token* de identidad quiere decir, que los servicios *cloud* disponen de la información del contexto del dispositivo IoT que lo ubica dentro del mundo físico gracias a las características de dicho contexto. Como esta información es almacenada en el *cloud*, este flujo de actuación se plantea mediante la técnica de direccionamiento inverso, es decir, en lugar de tener el servicio *cloud* que llegar uno a uno a cada uno de los dispositivos IoT para que realicen una u otra acción, es el propio servicio *cloud* quien genera y almacena el conjunto de eventos de actuación gracias a la información de contexto de los diferentes dispositivos IoT y son éstos quién acceden a dicho *cloud* para consultar qué tienen que hacer cada cierto tiempo.

Esta clase de direccionamiento se basa en los mecanismos de las arquitecturas dirigidas por eventos (EDA - *Event-Driven Architecture*), las cuales permiten una construcción de grandes aplicaciones distribuidas y altamente escalables. La ca-

racterística principal que permite este tipo de soluciones tan altamente escalables se apoya en los mecanismos de publicación-suscripción en lugar de comunicaciones punto a punto síncronas. Por un lado, la publicación del evento con la acción que se requiere que realice el dispositivo IoT por parte del servicio *cloud* se lleva a cabo en un sistema centralizado, lo que permite a dichos servicios *cloud* independizarse de los protocolos de comunicación subyacentes de dichos dispositivos y de sus características técnicas de bajo nivel. Por otro lado, los dispositivos IoT, al ser ellos quién se suscriben a un evento u otro para recibir las diferentes acciones que deben realizar a partir de las decisiones de los servicios *cloud*, permiten al sistema evitar la necesidad de tener un traductor de nombres que mapee el nombre del dispositivo a la dirección lógica o dirección física del dispositivo y la correspondiente tabla de enrutado para poder remitirle la acción correspondiente desde el *cloud*. Esto es así ya que es el propio dispositivo es el que se encarga de acceder al sistema centralizado en el que se publican los eventos para recoger el suyo, lo cual proporciona al dominio de aplicación en el que se despliegue este modelo de direccionamiento inverso, entre otros, una mejor calidad del servicio asegurando qué dispositivos IoT han obtenido las acciones a realizar y una mejor trazabilidad en todo momento al saber qué dispositivo IoT está accediendo a leer qué acción y si realmente tiene autorización a realizar dicha lectura.

Para realizar un uso óptimo de los recursos del dispositivo IoT y del consumo que éste realiza de su batería, los dispositivos IoT acceden a dicho sistema centralizado en el que se publican los eventos a través de sus correspondientes dispositivos *edge*, lo que les permite no tener que disponer de características de seguridad extra para llevar a cabo una autenticación robusta contra dicho sistema de publicación de eventos. Por otro lado, al estar apantallados por los dispositivos *edge*, los dispositivos IoT realizan consultas recurrentes para conocer si tienen o no una nueva acción que realizar para modificar su estado y por ende, el del mundo físico que les rodea. La periodicidad entre consulta y consulta puede ser configurada en función de la propia periodicidad de la publicación de eventos llevada a cabo por los servicios *cloud* o incluso, por las necesidades de los dispositivos IoT dentro de su dominio de aplicación en el que están desplegados. En la figura 3.6 se muestra el flujo de actuación propuesto que soporta el direccionamiento inverso propuesto. Dicho flujo se compone de los siguientes pasos:

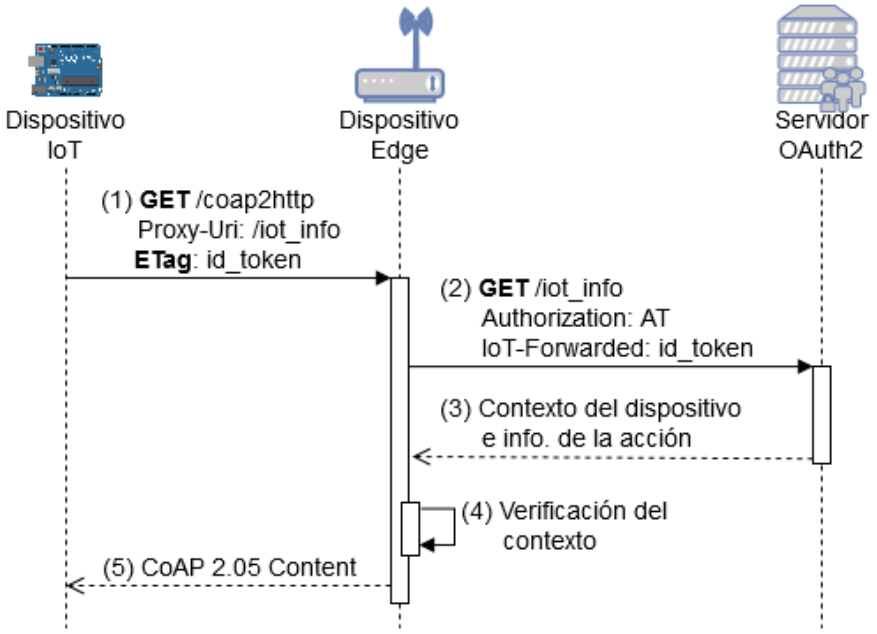

Figura 3.6: Diagrama de secuencia del flujo de actuación

1. Del mismo modo que se ha descrito anteriormente para el flujo de acceso, el dispositivo IoT envía en su petición al dispositivo *edge* dos cabeceras incluidas dentro de la propia especificación del protocolo CoAP: *ETag* y *Proxy-Uri*. Dentro de la cabecera *ETag* se incluye el *token* de identidad del dispositivo IoT que obtuvo tras realizar su registro en el sistema. Por otro lado, para este caso concreto del flujo de actuación, en la cabecera *Proxy-Uri* se envía siempre la ruta específica del servidor de OAuth 2.0 en la cual un dispositivo IoT puede consultar a través de su dispositivo *edge* la información asociada a su propio contexto entre la que se incluye, entre otros, el estado o acción que el servicio *cloud* requiere que tenga dicho dispositivo en cada momento.

2. El dispositivo *edge* realiza la traducción de la petición CoAP recibida a su homóloga HTTP y la envía a la URI indicada en la cabecera *Proxy-Uri*. En esta solicitud, el servidor de OAuth 2.0 recibe tanto el *token* de identidad del dispositivo IoT como el *token* de acceso del dispositivo *edge*. En el ejemplo mostrado en la figura 3.6, el método GET de CoAP es traducido

al correspondiente método GET de HTTP.

3. El servidor de OAuth 2.0 recupera la información del contexto asociada al *token* de identidad, incluyendo la acción u orden que se requiere que sea ejecutada por el dispositivo IoT. Dicha acción u orden ha sido previamente cargada por el servicio *cloud*. A continuación, dicha información se envía de vuelta al dispositivo *edge*.

4. El dispositivo *edge* verifica que el contexto del dispositivo IoT recuperado de la petición inicial y el obtenido a partir de la consulta al servidor de OAuth 2.0 coinciden. Si dicho contexto no coincide, la información relativa a la acción u orden a realizar recuperada de dicho servidor no será compartida con el dispositivo IoT.

5. Finalmente, si la verificación del contexto se llevó a cabo correctamente, el dispositivo *edge* traduce nuevamente la respuesta HTTP a su homóloga CoAP de tal forma que incluya en ella la acción u orden solicitada requerida por el servicio *cloud*. En este caso, como la verificación del contexto ya se ha llevado a cabo por parte del dispositivo *edge*, sólo sería necesario retornar al dispositivo IoT la acción u orden en sí, ya que no aportaría nada proporcionarle la misma información de contexto que ya dispone el dispositivo IoT al propio dispositivo.

3.5.2. Esquema de nombrado

Dado que la arquitectura de referencia mostrada en la figura 3.1 sigue una estructura jerárquica, el esquema de nombrado obviamente debería seguir una estructura jerárquica también. Dicha estructura de nombrado debe ser extensible para cualquier dominio de aplicación, ecosistema, propietario o proveedor de sistemas englobados dentro del Internet de las cosas ya que esto facilitaría la compartición de información, aunque dada la heterogeneidad de este tipo de ecosistemas, es poco probable que todos los servicios *cloud*, IoT y los propios dispositivos *edge* utilicen un esquema de nomenclatura común y único. Sin embargo, hay que tener en cuenta que el nombrado de un dispositivo puede ser visto como su identificador unívoco dentro de su dominio de aplicación, incluso si este se mueve dentro

de la red o se va registrando en distintos dispositivos edge. Esta clase de nombrado puede ser usado, por ejemplo, para verificar la integridad u origen de los datos, para garantizar el no repudio o para ejecutar búsquedas o descubrimientos de dispositivos basados en su nombre dentro del sistema.

El uso de un nombrado jerárquico minimiza la probabilidad de colisión en los nombres haciendo más fácil comprobar la veracidad y unicidad de un nombre dentro de un sistema así como la realización del mapeo de dicho nombre con el dispositivo específico que representa dentro de un gran despliegue. Obviamente, esta capacidad de identificar dispositivos en un sistema altamente escalable puede ser mejorado añadiendo incluso etiquetas o campos extras basados en las características de dichos dispositivos con el objetivo de que el sistema pueda realizar tareas de agrupación y agregación de nombres en base, por ejemplo, a funcionalidades soportadas. Además, un esquema jerárquico proporciona una buena compatibilidad con los sistemas de nombrados existentes en Internet a día de hoy como puede ser, por ejemplo, los servicios DNS [168] (*Domain Name System*). Otro esquema de nombres jerárquico ampliamente extendido es el basado en el estándar XRI [169] (*eXtensible Resource Identifier*). Este estándar proporciona un formato de nombre entendible tanto por humanos como por máquinas basado en etiquetas [170] que puede ser expresado como URIs si fuera necesario, o incluso, ser persistido sin problemas.

El esquema de nombres propuesto en esta tesis funciona de manera muy acoplada con el direccionamiento dirigido por eventos descrito en esta sección. Por un lado, el esquema de nombrado se implementa gracias al flujo de registro, y por otro, el modelo de direccionamiento se apoya, como ya se ha comentado, en la arquitectura de referencia propuesta, incluyendo el servidor de autorización de OAuth 2.0. La relación entre ambos puntos permite conocer la localización de cada uno de los dispositivos IoT registrados en el sistema en cada momento o incluso, las áreas en las que existe una mayor densidad de dispositivos desplegados dentro del sistema desde un punto de vista lógico sin tener en cuenta los detalles de despliegue de red o los protocolos de comunicación subyacentes.

Por otro lado, la vinculación entre OAuth 2.0 y el esquema de nombres propuesto permite la capacidad de auto configuración dentro del flujo de registro en el sistema. Esta configuración automatizada es responsabilidad del servidor de

OAuth 2.0, el cual no sólo proporciona resolución de nombres sino también permite prevenir la suplantación de identidad de los diferentes dispositivos gracias a su identificación y contexto. En este caso, una suplantación de identidad de un dispositivo IoT podría ser detectado gracias a la propia colisión de nombres dentro del esquema y en tal caso, el propio servidor de OAuth 2.0 podría expulsar al dispositivo malicioso del sistema revocando su nombre y su *token* de identidad o, en el caso de que la suplantación afectará a más dispositivos dentro de alcance del mismo dispositivo *edge*, como ya se ha comentado, revocar el *token* de acceso de dicho dispositivo de borde.

La materialización del esquema de nombres propuesto en esta tesis se basa en el estándar XRI gracias al cual, el nombre o identificador en sí que se genera para un dispositivo IoT registrado en el sistema puede construirse basado en cinco etiquetas:

xri:// Dominio aplicación / Región / Zona / Dispositivo Edge / Dispositivo IoT

Con respecto a las características de cada una de las distintas etiquetas que conforman la XRI, a continuación se detallan sus características:

- Dominio de aplicación: la premisa inicial dentro de este esquema de nombrado es que todo nombre o identificador de cualquier dispositivo debe ser agnóstico de su proveedor de identidad o servidor de OAuth. Debido a esto, el nombre debe ser único e interoperable para cada dispositivo IoT entre los distintos sistemas dentro del mismo dominio de aplicación. En este caso, esta etiqueta comienza con un ”=” o un ”+” dependiendo si es un dominio de aplicación de cualquier clase de negocio IoT o una persona respectivamente.

- Región y Zona: son la segunda y tercera etiqueta respectivamente dentro del XRI. Estas etiquetas representan como los dispositivos IoT se agrupan de una manera geográfica u organizacionalmente. Ambas son etiquetas opcionales y pueden ser omitidas si el despliegue del sistema no es muy complejo aunque pueda ser un despliegue global. Dichas etiquetas comienzan

por "@" porque ambas están relacionadas con datos de geolocalización o unidades organizativas.

- Dispositivo *Edge*: esta etiqueta se usa para identificar el dispositivo *edge* o de borde dentro del dominio de aplicación, región y zona específica que está conectado directamente con los dispositivos IoT, el cual puede ser, entre otros, un gateway, un controlador embebido o independiente, o incluso una conjunción de dispositivos edge que forman un cluster o un centro de procesamiento de datos micro. Esta etiqueta es la primera dentro de la jerarquía de nombrado que está relacionada con un aspecto tecnológico dentro del mundo físico y no funcional, y por tanto, comienza con "+" porque puede ser cualquier tipo de los dispositivos indicados.

- Dispositivo IoT: esta etiqueta representa cualquier tipo de dispositivo embebido dentro del mundo físico, con capacidades y recursos limitados de cómputo, almacenamiento u otros. Por ejemplo, estos dispositivos pueden ser sensores de cualquier tipo como de temperatura, de humedad, etc., o incluso actuadores que interactúan con el mundo físico. Por tanto, esta etiqueta, del mismo modo que la anterior, comienza por "+".

Si tomamos como ejemplo el dominio de aplicación de la agricultura inteligente focalizado en la ganadería vacuna con un alcance de proyecto de nivel europeo, un ejemplo de nombre de un dispositivo IoT que permitiría monitorizar a una de las vacas de una determinada ganadería en concreto sería el siguiente:

xri:// =SmartFarming / @Spain / @Madrid / +FarmEdgeDevice1 / +CowSensor1

Como se puede apreciar, la estructura del nombre XRI del dispositivo IoT se apoya en la estructura jerárquica mostrada en la figura 3.7, donde el dominio de aplicación es la propia agricultura inteligente (*Smart Farming*), la región, al ser un proyecto a nivel europeo, sería el país (*Spain*), y la zona, la provincia concreta (Madrid). La siguiente etiqueta indica el dispositivo edge concreto a través del cual se ha registrado el dispositivo IoT dentro de la granja, en la cual podría haber

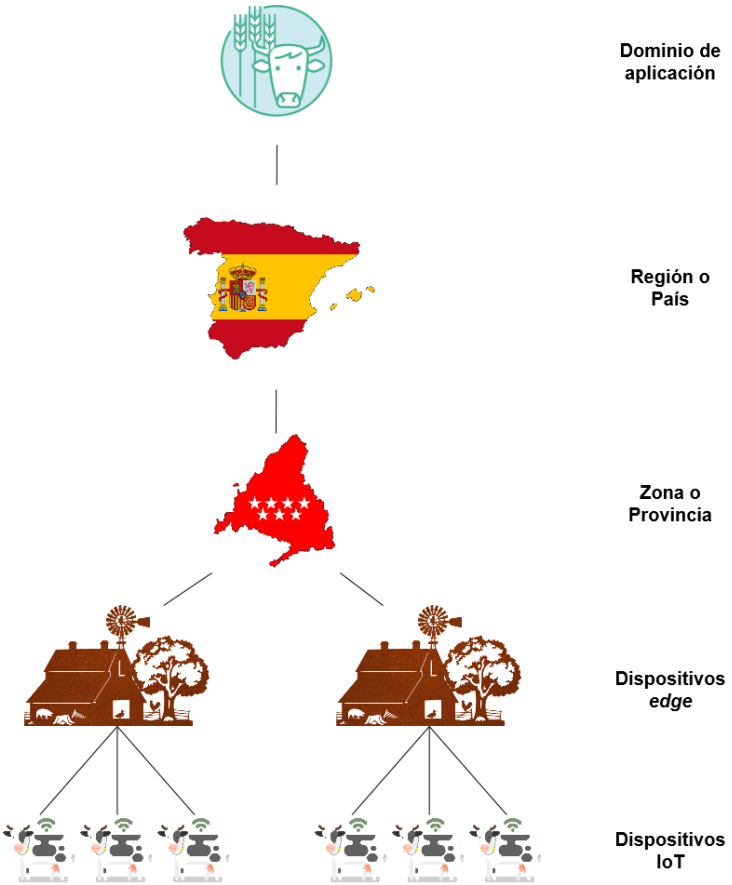

Figura 3.7: Ejemplo de la estructura jerárquica de nombrado definida

uno o varios dispositivos *edge* en función del tamaño de la misma. La última etiqueta indica el sensor concreto de la vaca en cuestión que la ubica unívocamente dentro de toda esta jerarquía.

Finalmente, como ya se ha comentado, no todos las etiquetas serían obligatorias dentro del nombre en formato XRI. Más concretamente, si se estuviera abordando un proyecto de dispositivos *wearables* con personas, las etiquetas de región y zona no serían necesarias ya que el propio dominio de aplicación es una persona concreta. En este caso, un ejemplo de nombre concreto sería tan sencillo como el que se muestra a continuación:

xri:// =Persona1 / +EdgeDevice1 / +Wearable1

Esta estructura de nombrado jerárquico, al ser gestionada por los propios servicios *cloud*, encaja perfectamente con el direccionamiento basado por eventos planteado en el flujo de actuación descrito en esta sección. Esto es así ya que cuando el propio servicio *cloud* quiere trasladar un conjunto de acciones u órdenes a realizar por parte de los diferentes actuadores, sólo tiene que utilizar la jerarquía de nombrado para especificar qué dispositivos tienen que cambiar el estado del mundo físico en el que están desplegados. De esta forma, una orden puede ser enviada a un sensor en concreto, a todos los sensores dependientes sólo de un dispositivo *edge*, o incluso, a nivel de zona o de región. Esta característica permite aprovecharse de las capacidades de caché de los diferentes dispositivos disponibles entre el servicio *cloud* y los dispositivos IoT con el fin de evitar la saturación de la red y mejorar el rendimiento del sistema.

Capítulo 4

Crowdsensing influenciado en entornos multidominio

Abordados los problemas principales de la gestión de identidades y accesos, y del direccionamiento y del nombrado derivados de la alta escalabilidad de los entornos del Internet de las cosas, el siguiente aspecto clave que conviene resolver es el de la interoperabilidad entre diferentes dominios de aplicación que por su naturaleza deben trabajar de manera colaborativa para construir conocimiento preservando en la medida de lo posible la privacidad de los individuos que se ven involucrados de uno u otro modo en dichos dominios de aplicación. Por este motivo, en este capítulo se desarrollan mecanismos para realizar un mejor aprovechamiento del conocimiento colectivo que proporcionan los diferentes dispositivos IoT desplegados en cada uno de los dominios de aplicación que trabajan de manera colaborativa entre sí. Para ello, se avanza el esquema de delegación de autorización propuesto en el capítulo anterior para ser utilizado en esquemas de *crowdsensing* en entornos multidominio.

4.1. Motivación

La necesidad de monitorizar entornos cada vez más complejos en tiempo real con el fin de proporcionar a los diferentes usuarios una experiencia enriquecida y personalizada provoca que la cantidad de sensores que recaban datos del mundo

físico en el que están desplegados sea enorme. Dicha experiencia personalizada es la que obliga, en muchos casos, a recabar información (comportamientos, hábitos, rutinas, preferencias) de cada uno de los diferentes usuarios, lo que puede implicar una amenaza para su privacidad.

Existen casos de uso en los que el coste derivado del despliegue de diferentes dispositivos IoT en un dominio de aplicación es tan grande que deriva en el empleo de esquemas *mobile crowdsensing* más económicos mediante los que los usuarios facilitan el uso de los sensores de sus propios dispositivos para enriquecer la información recabada por los diferentes dominios de aplicación sacrificando parte de su privacidad en aras de percibir como contrapartida una recompensa o incentivo económico proporcional a la cantidad y calidad de los datos facilitados. Es esta última característica la que impide que los datos recabados a través de estos esquemas sean completamente anonimizados ya que en ese caso, no podría materializarse el pago de las contraprestaciones de manera personalizada. Por este motivo, se hace necesario en estos esquemas el uso de técnicas criptográficas y de ofuscación para preservar esa privacidad individual, lo que repercute en un coste computacional considerable por el mero hecho de utilizar sensores de diferentes dominios de aplicación para enriquecer la información de los servicios *cloud* de uno de ellos en concreto en lugar de que los servicios *cloud* de ambos dominios de aplicación colaboren entre sí. Buenos ejemplos de dominios de aplicación en los que encontramos esta casuística concreta serían las ciudades inteligentes y las carreteras inteligentes.

Las limitaciones y aspectos en común que encontramos en cada uno de los diferentes casos de uso mencionados que intentan abordar los trabajos analizados en el capítulo 2 de estado del arte se pueden resumir en los siguientes puntos:

- El requisito de respuesta en tiempo real que se le exige a día de hoy a los diferentes entornos en los que se despliegan los dispositivos IoT que proporcionan una experiencia enriquecida al usuario está provocando la rápida adopción de los enfoques basados en arquitecturas *edge-centric*. Esta tendencia en el estado del arte del *crowdsensing* está potenciada por la mejora en los tiempos de latencia en las comunicaciones y la capacidad de cómputo que proporcionan los dispositivos *edge* al ecosistema del Internet de las co-

sas al estar desplegados tan próximos de los diferentes sensores y actuadores empotrados en el mundo físico. Por lo tanto, el esquema *edge-centric* propuesto en el capítulo anterior parece un buen punto de partida para proponer mecanismos de *crowdsensing* en el presente capítulo, está en la misma línea que los últimos trabajos de investigación del área.

- Los problemas de privacidad identificados hasta el momento en los trabajos de *crowdsensing* fomentan el uso de técnicas criptográficas para preservar la privacidad de los usuarios (sobre todo en los ecosistemas de *mobile crowdsensing*), lo que repercute de manera negativa en las arquitecturas *edge-centric* y dispositivos IoT ya que fuerza a dichos dispositivos a realizar cálculos computacionalmente costosos para los que su hardware no está preparado dadas sus limitaciones intrínsecas. Además, la capacidad de obtener información enriquecida en estos esquemas depende de la voluntariedad de los usuarios de formar parte de ellos ya sea por interés propio o por el incentivo económico en sí, ya que son los usuarios los que sacrifican parte de su privacidad en beneficio del dominio de aplicación. Esto provoca un sesgo bastante importante en la obtención de información ya que siempre dicha recolección estará supeditada a la voluntad de los propios usuarios para tener disponibles en sus dispositivos las aplicaciones necesarias para el dominio de aplicación concreto.

- Los trabajos previos en el ámbito del *crowdsensing* plantean contextos de *clouds* no colaborativos entre distintos dominios de aplicación, es decir, suelen basarse en arquitecturas de referencia en las que existe un único servicio *cloud* que, para poder monitorizar o controlar su dominio de aplicación, necesita recabar información del máximo número de dispositivos disponibles aunque éstos formen parte de otro dominio de aplicación y persigan otro fin. En este trabajo se pretende cambiar el enfoque, de manera que los servicios *cloud* asociados a los diferentes dominios de aplicación sigan sin colaborar explícitamente entre ellos para garantizar la privacidad de los usuarios involucrados pero se puedan influenciar los unos a los otros mediante sus decisiones. La idea fundamental es que un usuario sólo comparta su información con el servicio de su dominio de aplicación (en el que confía), con

las suficiente garantías de privacidad como para que no renuncie a este aspecto por participar en la aplicación de *crowdsensing*. Se parte de la base de que un dominio de aplicación es capaz de, a partir del estudio del entorno en el que están desplegados sus sensores, recabar la información suficiente para tomar decisiones e influir a través de sus actuadores en ese mismo entorno. El resto de dominios de aplicación puedes percibir esas modificaciones y adaptarse o actuar nuevamente sobre el entorno, de tal forma que todos dominios busquen el mejor estado de equilibrio posible para el contexto en el que están desplegados sin necesidad de compartir información de índole privada de los usuarios entre ellos.

4.2. Arquitectura de referencia y asunciones

Como se puede observar en la figura 4.1, la arquitectura de referencia establecida se basa en la existencia de múltiples dominios de aplicación que siguen el enfoque *edge-centric* descrito en el capítulo 3. La diferencia existente en dicha figura entre el denominado dominio de aplicación 1 y el resto de dominios de aplicación se debe a que es necesario elegir siempre un dominio de aplicación concreto de los involucrados en el esquema de *crowdsensing* influenciado como dominio de base o de referencia, el responsable de tomar las diferentes mediciones del entorno para influenciar al resto de dominios de aplicación, los cuales a su vez, pueden influenciar tanto al dominio base como al resto de dominios de aplicación hermanos. En base a esta descripción, la arquitectura de referencia define dos roles funcionales con las siguientes características:

- Dominio de aplicación base: es aquel que se toma de referencia para poder realizar las diferentes mediciones y captar las alteraciones físicas que el resto de dominios de aplicación generan en el entorno que se está monitorizando. Este dominio se despliega cubriendo una superficie geográfica extensa en el mundo físico, normalmente estática y bien acotada. De ahí que las diferentes alteraciones producidas por los diferentes dominios observados en el mundo físico se midan en base a las magnitudes de este dominio

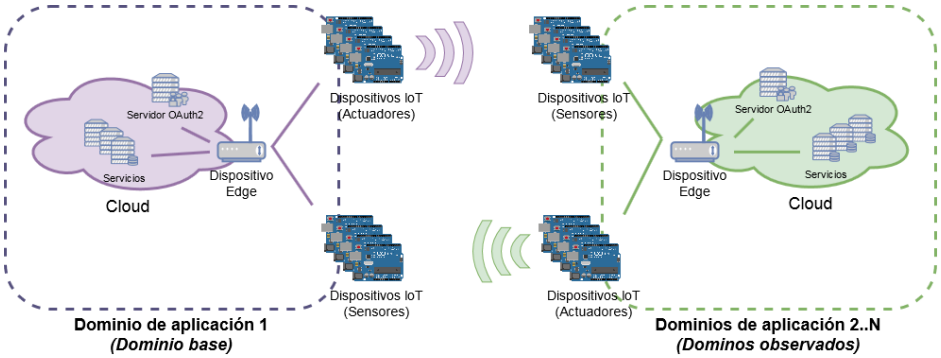

Figura 4.1: Arquitectura *crowdsensing* de referencia

de referencia.

- Dominio de aplicación observado: son todos aquellos dominios observados y monitorizados tanto por el dominio base como entre ellos. Pueden influenciar con su comportamiento tanto al dominio de aplicación base como al resto de sus dominios de aplicación hermanos. Suelen desplegarse en superficies geográficas reducidas, y presentar características de dinamismo, movilidad o fronteras borrosas.

Como se puede deducir de estas definiciones, los sensores disponibles en el dominio de aplicación base deben permitir la obtención de información del entorno y de las diferentes alteraciones que generan en él el resto de dominios, mientras que sus actuadores deben ser capaces de influenciar a todos los dominios de aplicación observados a la vez. Por otro lado, los sensores y actuadores de los dominios de aplicación observados deben cumplir una doble función. En primer lugar, sus sensores deben ser capaces de percibir las diferentes alteraciones que suceden en el mundo físico, incluyendo las provocadas por cualquiera de los dominios de aplicación desplegados en dicho entorno, es decir, tanto dominio base como el resto de dominios observados. Y en segundo lugar, sus actuadores deben poder realizar su función del mismo modo en el entorno, es decir, influir en todos los dominios de aplicación existentes en él. Este comportamiento descrito en el que el dominio base y los dominios observados se atraen e influencian mutuamente entre sí, y los dominios observados tienden a mantenerse en un entorno competitivo entre ellos,

añade a la arquitectura de referencia la característica de ser una arquitectura magnética, ya que existe una relación de atracción entre los dominios de aplicación estáticos (base) y los móviles (observados), mientras que entre dominios móviles, se repelen.

Para finalizar la descripción de la arquitectura de referencia que se va a tener en cuenta para proponer los mecanismos de *crowdsensing* respetuosos con la privacidad en este capítulo, es necesario completar lo discutido hasta este punto con algunas particularidades técnicas:

- Los sensores y actuadores desplegados tanto por el dominio de aplicación base como por los dominios de aplicación observados deben soportar el uso del protocolo CoAP sobre DTLS en sus comunicaciones con los dispositivos *edge*. De esta forma, se podrá confiar en el esquema de gestión de identidades y en el modelo de direccionamiento y nombrado propuestos en el capítulo anterior de este documento.

- Con el fin de preservar la privacidad de datos como la geolocalización de los usuarios o de los dispositivos, cada dispositivo *edge* desplegado, además de soportar OAuth 2.0 y HTTPS, se debe ubicar en el mundo físico de tal forma que agrupe bajo su control a un número similar de sensores y actuadores al del resto de dispositivos de borde. De esta forma se evitan desequilibrios que faciliten a un tercero extrapolar los datos de geolocalización del resto de actores vinculándolos a dominios de aplicación concretos.

- Debe existir una segmentación de red en cada dispositivo *edge* que permita separar en un segmento diferenciado a los sensores y en otro a los actuadores con el fin de gestionar de una manera óptima las comunicaciones con cada una de ellos, incluyendo las comunicaciones *multicast*.

- Por último, los dispositivos *edge* del dominio base deben disponer de un segmento de red independiente al de sus sensores y al de sus actuadores con el objetivo de que dicho segmento sea el utilizado por los diferentes dispositivos *edge* de los dominios observados para enriquecer la información recopilada por el sistema base sin comprometer los dispositivos IoT des-

plegados por el dominio base ni corromper la información gestionada entre ellos y su dispositivo *edge* correspondiente.

4.3. Paradigmas y especificaciones fundamentales

Al contrario que en el capítulo 3, el pilar fundamental del esquema de *crowdsensing* influenciado propuesto no es ningún aspecto tecnológico como un protocolo de comunicación o un *framework* de autorización, sino más bien, la forma de modelar el comportamiento del sistema en su conjunto. Una de las premisas establecidas en la arquitectura de referencia definida en este capítulo es la existencia de un dominio de aplicación base que recoge información de los dominios de aplicación que están siendo observados y con este conocimiento colectivo, produce algún tipo de decisión o *feedback* para influir sobre ellos. Por ello se considera que un paradigma básico que se puede tener en cuenta para proponer los mecanismos de *crowdsensing* objeto de esta investigación son los modelos de teoría de colas [171].

Esta teoría describe cómo un sistema que dispone de una serie de recursos limitados atiende las diferentes peticiones de una población de clientes y establece, en caso de no haber disponible ningún recurso en el sistema en ese momento, una cola o línea de espera para los peticionarios hasta que puedan ser atendidos. Esta definición encaja perfectamente con la arquitectura de referencia utilizada ya que en ella, el sistema que proporciona los recursos es el dominio de aplicación base y la población de clientes que necesitan consumir sus servicios son el resto de dominios de aplicación, los observados. Este tipo de modelo es lo suficientemente genérico para capturar el comportamiento de una gran cantidad de dominios de aplicación diferentes y de casos de uso, ya sea a pequeña escala como por ejemplo, como edificios, granjas o gasolineras, como a gran escala en ciudades o carreteras.

Cabe destacar también que este tipo de modelo permite un análisis de los comportamientos y tendencias en los dominios de aplicación base tanto por parte de sus servicios *cloud* (que disponen de la muestra completa en cada momento del conjunto de datos global), como por parte de sus dispositivos *edge* (los cuales disponen de una muestra menor que se adecua a la capacidad de cómputo que poseen dichos dispositivos y que se obtiene a partir exclusivamente de los sen-

sores de los que son responsables directamente). Este tipo de análisis por niveles permite influir en el mundo físico sin necesidad de preocuparse de identidades de dispositivos concretos o de analizar información privada de los usuarios involucrados.

En resumen, en este trabajo se propone utilizar como base de los mecanismos de *crowdsensing* para la arquitectura de referencia que se muestra en la figura 4.1 la teoría de colas, empleando en principio modelos G/G/n/m, lo que significa que:

- Se supone una distribución general (G) para los tiempos entre llegadas de los clientes (peticiones de servicio por parte de los dominios observados), ya que dependerán del caso de uso concreto.
- Se supone también una distribución general (G) para el tiempo de servicio en el dominio de aplicación base (de nuevo por la dependencia del caso de uso).
- Se suponen n servidores en el dominio de aplicación base, este factor de nuevo dependerá del caso de uso concreto.
- La capacidad máxima es m, es decir, se pueden gestionar m dominios observados por servidor, el $m+1$ se rechazaría y no se podría poner en cola para ser atendido o procesado por un servidor en el dominio base. Este factor es de nuevo muy dependiente del caso de uso.
- La disciplina de cola es FIFO (First In, First Out), es decir, los trabajos en cola son servidos en riguroso orden de llegada.

4.4. Solución propuesta para el conocimiento colectivo y respetuoso con la privacidad en IoT

En esta sección se muestra cómo el dominio de aplicación base del mecanismo de *crowdsensing* influenciado propuesto es capaz de analizar en sus dispositivos *edge* los diferentes comportamientos captados a través de sus sensores para proporcionar una experiencia enriquecida y en tiempo real al resto de dominios de

aplicación observados como consecuencia de las decisiones que tome gracias al conocimiento colectivo y a la teoría de colas. Estas decisiones son las que provocan que dicho dominio base pueda influir en el resto de dominios de aplicación a través de sus diferentes actuadores alterando el estado del mundo físico, y que sean dichas alteraciones percibidas por los dominios observados las que provoquen la adecuación de éstos a las nuevas condiciones del entorno. Obviamente, cada alteración y cada obtención de información se realiza a través de dispositivos IoT que minimizan la recolección de datos de índole personal y que evitan la compartición de identidades entre los diferentes dominios de aplicación involucrados gracias al enfoque *edge-centric* seguido en esta tesis doctoral.

En la figura 4.2 se muestra el flujo propuesto. En él se identifica, por un lado, a los cuatro actores involucrados por parte del dominio base: un sensor y un actuador genéricos, un dispositivo *edge* que se encargará de los cálculos y calibraciones empleando teoría de colas gracias a la información recabada por los sensores de los que es responsable, y los propios servicios *cloud* del dominio de aplicación base que dispondrán en todo momento de la información global del sistema. En dicho flujo se puede observar que, la obtención de información a través del sensor es la responsable del inicio del flujo, y una vez analizados los datos obtenidos por parte del dispositivo *edge*, éste se encarga de adecuar el estado del entorno gracias al actuador. Por otro lado, se identifica también el dispositivo *edge* de un dominio observado que participará en la recolección de información para enriquecer los cálculos y calibraciones del dispositivo *edge* del dominio base a partir de información recogida de sus propios sensores. Por simplicidad, se han omitido en el flujo los sensores y actuadores bajo la responsabilidad de este dispositivo *edge* del dominio observado aunque se presupone su existencia (como se mostró en la arquitectura de referencia de la sección 4.2).

En primer lugar, dentro del flujo se identifica un acceso a los servicios *cloud* para la creación de un recurso que se basa en la información recabada por el sensor. Esta parte del flujo (del paso 1 al paso 5) sería una versión simplificada del flujo de acceso a recursos ya descrito en el capítulo 3. LEn este caso concreto el dispositivo *edge*, además de actuar como traductor del protocolo de mensajería y *proxy* de características de seguridad, también debe retener la información más relevante del contexto que le ayude en sus análisis y cálculos en los siguientes

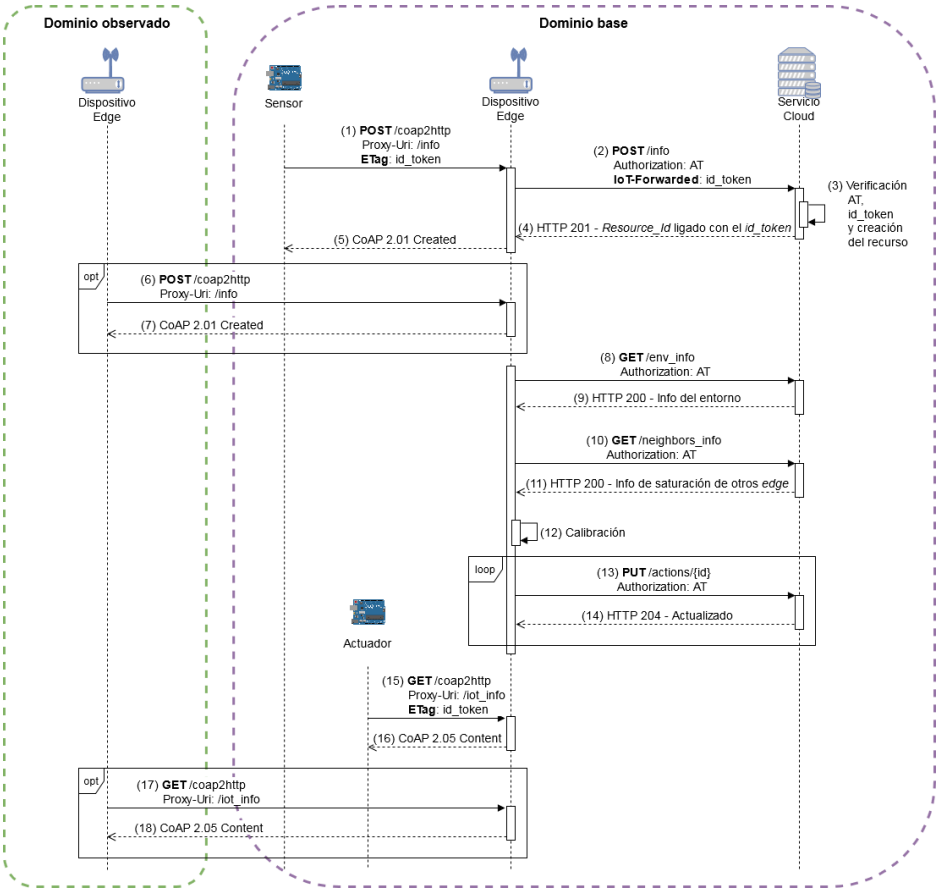

Figura 4.2: Diagrama de secuencia del flujo para el *crowdsensing* influenciado

pasos del flujo.

En segunda instancia, y con carácter opcional en función de la predisposición de los usuarios a compartir información, los dispositivos *edge* de los diferentes dominios observados pueden iniciar una comunicación con el dispositivo *edge* del dominio base a través del segmento de red dedicado que éste dispone para remitirle diferentes datos de telemetría del dominio observado que puedan ayudarle en las calibraciones posteriores. Es importante observar que en los pasos 6 y 7 del flujo no se proporciona ningún tipo de *token* de identidad, ni se envían al *cloud* los datos recabados, ya que sirven como información de enriquecimiento para los cálculos posteriores. No será necesario compartir datos sensibles como la geolocalización de los dispositivos, ya que la propia ubicación del dispositivo *edge* del

dominio base proporciona la ubicación aproximada (o relativa, si fuera necesario medir distancias) de los diferentes dominios observados sin necesidad de que el dato exacto sea compartido entre ambos dominios.

En tercer lugar, el dispositivo *edge* se activa de manera autónoma con cierta periodicidad con el fin de recabar en el paso 8 información del entorno y de sus propios servicios *cloud* como puedan ser características físicas (lo que puede influir en el número de servidores del modelo de colas que aplica), capacidades técnicas (lo que puede influir en los tiempos de respuesta) o constantes que definen limitaciones regulatorias que debe tener en consideración a la hora de realizar sus cálculos y análisis sobre el comportamiento de los diferentes usuarios que están interactuando con el entorno monitorizado desde cada uno de sus dominios de aplicación observados. Además, dicho dispositivo solicita en el paso 10 a sus servicios *cloud* la información relativa al estado de los dispositivos *edge* más próximos a él dentro del dominio base con el fin de poder identificar situaciones de congestión o saturación en el uso de recursos de una zona contigua. De esta forma se pueden tomar decisiones no sólo basadas en la propia información que posee el dispositivo *edge*, sino enriquecidas gracias a las tendencias de comportamiento identificadas por sus compañeros más cercanos.

Una vez dispone de toda la información necesaria, el dispositivo *edge* realiza en el paso 12 los análisis necesarios para identificar cómo mejorar el estado del dominio de aplicación base en su conjunto gracias al cálculo del nivel de saturación de los recursos bajo su responsabilidad con un modelo basado en teoría de colas. Teniendo en cuenta que tanto el número de servidores como el ratio medio de servicio de los mismos y el factor de utilización máximo del sistema son datos de los que el dispositivo *edge* dispone a partir del paso 8 cuando consulta al servicio *cloud*, en función del dominio de aplicación concreto, el dispositivo *edge* puede, a partir del conjunto de observaciones obtenidas a través de sus sensores y de los dispositivos *edge* de los dominios observados, establecer espacios temporales en los que calcular el ratio medio de llegada de peticiones real para compararlo con el estimado que obtiene a partir de los datos proporcionados por el *cloud*, ya que dada su naturaleza dicho dispositivo no dispone de toda la información del comportamiento de los usuarios en el sistema o de la capacidad global en tiempo real del mismo. Esto le permite concluir su análisis con la necesidad de mantener

su estado actual o con la necesidad de atraer a su zona a más usuarios influyendo en el entorno a través de sus actuadores con el fin de liberar de carga a los dispositivos *edge* más próximos y que los usuarios puedan utilizar sus propios recursos disponibles. Finalmente, para mantener la consistencia del sistema, el dispositivo *edge* actualiza a los servicios *cloud* en el paso 13 el estado o acciones de los dispositivos IoT que va a modificar.

Manteniendo el flujo de actuación ya descrito en el capítulo 3, cuando el actuador del dominio base requiere información sobre lo que debe hacer, consulta al dispositivo *edge* en el paso 15. Éste le devolverá el estado o acción que calculó y que notificó al servicio *cloud* de tal forma que a partir de dicho momento, el sistema en su conjunto vuelve a mantener su consistencia en los tres niveles, es decir, mundo físico, dispositivos *edge* y *cloud*. Son en este momento los actuadores los que pasan a influenciar con su nuevo estado al resto de dominios de aplicación observados para que dichos dominios se adecuen al estado actual del entorno y a su vez, las actuaciones para adecuarse por parte de estos dominios de aplicación puedan volver a ser detectadas por los sensores del dominio de aplicación base sin necesidad de compartir ningún tipo de identidad de los usuarios en todo el flujo.

Para terminar, y con el mismo carácter opcional que los pasos 6 y 7 del flujo, en el paso 17 el dispositivo *edge* del dominio observado puede consultar las acciones o recomendaciones al dispositivo *edge* del dominio base de tal forma que pueda ser él el que modifique sus propios actuadores y calibre su propio dominio observado en base a la información obtenida en el paso 18 y a la que recaba a través de sus propios sensores.

Capítulo 5

Implementación, validación, análisis de seguridad y privacidad

A lo largo de este capítulo se aborda la implementación tanto de la solución de control de acceso, direccionamiento y nombrado propuesta en el capítulo 3 como de la solución de *crowdsensing* influenciado del capítulo 4. Se incluye también la validación de ambas soluciones en casos de uso reales y se presentan las pruebas de rendimiento llevadas a cabo. Finalmente, se discuten los análisis de seguridad y privacidad realizados para evaluar los niveles conseguidos por las soluciones propuestas.

5.1. Implementación del esquema propuesto

Teniendo en cuenta las tres capas consideradas en la arquitectura de referencia mostrada en la figura 3.1 del capítulo 3, se puede identificar que existen dos roles clave a la hora de dar soporte a la solución definida y sobre los cuales hay que llevar a cabo los desarrollos necesarios. Por un lado, el rol *cloud*, que incluye el servidor de autorización de OAuth 2.0, y por otro, el rol *edge*, que es el que se encarga de servir de traductor entre los dos mundos, es decir, el del *cloud* y el del Internet de las cosas. A lo largo de esta sección se describe el conjunto de APIs definidas y los modelos de datos necesarios para dar soporte al esquema propuesto.

5.1.1. Rol *cloud*

Se considera rol *cloud* al conjunto de servidores centrales que incluyen tanto los servicios de recursos protegidos específicos de la aplicación o dominio de aplicación concreto, así como el servidor de autorización con los diferentes añadidos que han sido desarrollados sobre la especificación de OAuth 2.0. La premisa principal para desarrollar los diferentes añadidos sobre la especificación de OAuth 2.0 es que cualquier funcionalidad o verificación que se lleve a cabo debe extender dicho protocolo sin afectar o modificar su implementación estándar ya que de esta manera, la inclusión de la funcionalidad en cuestión puede ser añadida en las diferentes implementaciones desplegadas a día de hoy en Internet como un añadido más y por lo tanto, crecer y mantenerse sin afectar al resto del sistema ya implantado en las diferentes organizaciones. El código fuente con las diferentes funcionalidades correspondientes al rol *cloud* desarrolladas de una forma completamente agnóstica de las diferentes características de uno u otro dominio de aplicación está disponible en [172].

APIs definidas

Para llevar a cabo la delegación de autorización de un dispositivo *edge* a un dispositivo IoT, como ya se ha comentado, es necesario que el primero disponga de un *token* de acceso válido de OAuth 2.0 ya que la delegación de autorización se realiza en base a la autorización en curso. Por este motivo, las diferentes APIs añadidas al servidor de autorización de OAuth 2.0 requieren estar protegidas por el propio protocolo, es decir, es condición *sine qua non* que en cada petición que realice el dispositivo *edge* a dichas APIs proporcione su *token* de acceso en la cabecera HTTP *Authorization* correspondiente. En la figura 5.1 se muestra la descripción *swagger* de las APIs.

A excepción del API de delegación de autorización, es decir, la API que está definida con el método HTTP POST en la figura 5.1, las otras tres APIs requieren además de la cabecera HTTP *Authorization* con el *token* de acceso del dispositivo *edge*, la cabecera concreta definida para esta extensión en la que se traslada el *token* de identidad del dispositivo IoT correspondiente, es decir, la cabecera HTTP *IoT-Forwarded*.

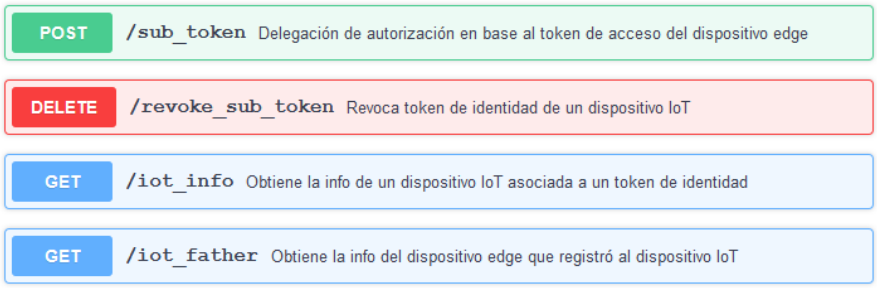

Figura 5.1: Extensión de APIs propuesta para el estándar OAuth 2.0

Figura 5.2: Definición del objeto JSON *IoT_Device*

Con respecto a cuáles son los parámetros de entrada y de salida de cada una de las APIs más allá de las cabeceras HTTP ya descritas, a continuación se describe la parametría concreta por cada una de las APIs:

- POST */sub_token*: para poder llevar a cabo la delegación de autorización del dispositivo *edge* al dispositivo IoT, el servidor de autorización de OAuth 2.0 debe recibir como parámetro, además de la cabecera HTTP *Authorization*, el objeto *IoT_Device* mostrado en la figura 5.2 que representa la información de contexto de dicho dispositivo IoT, y una vez procesada dicha petición, emitirá un *token* de identidad representado como el objeto *Id_Token* cuyos campos se muestran en la figura 5.3.

- DELETE */revoke_sub_token*: esta API no requiere de ningún campo ni de

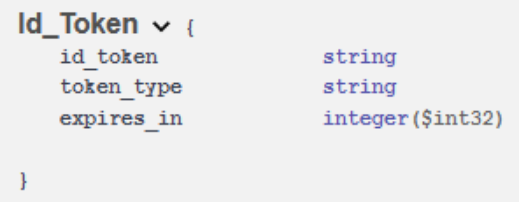

Figura 5.3: Definición del objeto JSON *Id_Token*

entrada ni de salida más allá de las dos cabeceras HTTP ya descritas. Una vez recibida la petición, el servidor de autorización revocará la autorización del dispositivo IoT representada mediante el *token* de identidad proporcionado por cabecera.

- GET */iot_info*: esta solicitud de información se lleva a cabo por el dispositivo *edge* cuando necesita validar la información de contexto del dispositivo IoT que le ha invocado y no dispone de ella en su almacenamiento local. Por este motivo, esta API recibe como parámetro las dos cabeceras HTTP ya descritas y devuelve el objeto *IoT_Device* mostrado en la figura 5.2 que representa la información de contexto de dicho dispositivo IoT.

- GET */iot_father*: esta API se utiliza cuando un dispositivo *edge* necesita saber qué otro dispositivo *edge* registró en el sistema al dispositivo IoT que le está invocando, posiblemente porque dicho dispositivo IoT esté en *roaming*. Para ello, el dispositivo que quiere realizar dicha petición invoca esta API con las dos cabeceras correspondientes y obtiene de vuelta la metainformación estándar del cliente de OAuth 2.0 que el propio servidor devolvería en su API estándar cuando se solicita el conjunto de información vinculada al *token* de acceso (también llamado introspección de *token*) por parte de la aplicación propietaria de dicho *token*.

Hasta este punto se ha definido el conjunto de APIs necesarias en el propio servidor de autorización de OAuth 2.0 para permitir extender su funcionalidad sin modificar o comprometer la propia del estándar, pero obviamente, se hace necesario definir cómo se materializa el control de acceso en el propio servidor de

```python
# ------------------------------------------------------------------
# ---- Testing APIs are implemented below
# ------------------------------------------------------------------


@bp.route('/events', methods=['POST'])
@require_oauth()
@require_oauth_iot()
def post_new_event():
    resource_id = g.pop('resource_id', None)
    # Resource creation code here ...
    response = __make_response(json.dumps({}, sort_keys=True), 201)
    response.headers["Location"] = "/events/" + resource_id
    return response


@bp.route('/events/<string:id>', methods=['PUT'])
@require_oauth()
@require_oauth_iot()
def put_event(id):
    # Resource modification code here ...
    return __make_response(json.dumps({}, sort_keys=True), 204)


@bp.route('/events/<string:id>', methods=['DELETE'])
@require_oauth()
@require_oauth_iot()
def delete_event(id):
    # Resource modification code here ...
    return __make_response(json.dumps({}, sort_keys=True), 204)


@bp.route('/events/<string:id>', methods=['GET'])
@require_oauth()
@require_oauth_iot()
def get_event(id):
    # Resource get code here ...
    return __make_response(json.dumps({}, sort_keys=True), 200)
```

Figura 5.4: Ejemplo de uso del decorador en python para proteger APIs

recursos protegido con OAuth 2.0 sin llegar a afectar a dicha especificación estándar. Para ello se ha llevado a cabo el desarrollo de un decorador que se encarga de realizar las validaciones correspondientes de tal forma que cuando el servidor de recursos necesita dar soporte al mecanismo de delegación de autorización propuesto, las APIs que se quieran proteger de esta manera deberían estar anotadas con dicho decorador como se muestra en el ejemplo de código de la figura 5.4.

El comportamiento del decorador a la hora de llevar a cabo las diferentes validaciones se puede resumir de la siguiente manera:

- Si la solicitud por parte del dispositivo *edge* o de cualquier otra aplicación tercera que no apantalle a ningún dispositivo IoT pero que esté incluida

dentro del sistema sólo incluye el *token* de acceso, el decorador asume que dicha petición no ha sido generada por un dispositivo IoT y por lo tanto, no realiza ningún tipo de validación de seguridad ya que se supone que la seguridad en el acceso al recurso será verificado dentro de los controles de acceso estándar de OAuth 2.0.

- Si la solicitud incluye tanto el *token* de acceso como el *token* de identidad, y además, el método HTTP utilizado es el método POST, el decorador verificará en primera instancia que el *token* de identidad no ha expirado ni ha sido revocado. Después de esto, el decorador verificará que existe una relación de delegación de autorización entre el *token* de acceso y el *token* de identidad proporcionados en la solicitud. En el caso de que estas validaciones previas se hayan llevado a cabo de manera satisfactoria, se generará un identificador de recurso ligado al *token* de identidad que se disponibiliza en el contexto del método HTTP POST para que identifique el recurso creado en dicha petición, siendo dicho identificador, el que será devuelto al dispositivo *edge*.

- En el caso de que la solicitud incluya tanto el *token* de acceso como el *token* de identidad, pero en este caso el método HTTP sea un GET, un PUT o un DELETE, además de llevar a cabo las mismas validaciones iniciales que en el caso anterior de validez del *token* de identidad y de la relación de delegación de autorización correspondiente entre ambos *tokens*, se llevará a cabo la verificación de que el identificador del recurso sobre el cual se quiere realizar el método HTTP correspondiente esté ligado a su vez al *token* de identidad incluido en la petición.

Modelo de datos y persistencia

El modelo de datos creado para dar soporte al rol *cloud* extiende el propio modelo de datos de OAuth 2.0 sin afectar o modificar su comportamiento estándar, es decir, sólo se añaden funcionalidades compatibles con el estándar no modificando las ya incluidas, lo cual mejora la capacidad de mantenimiento y gestión de la funcionalidad creada en esta tesis para el Internet de las cosas. En la figura

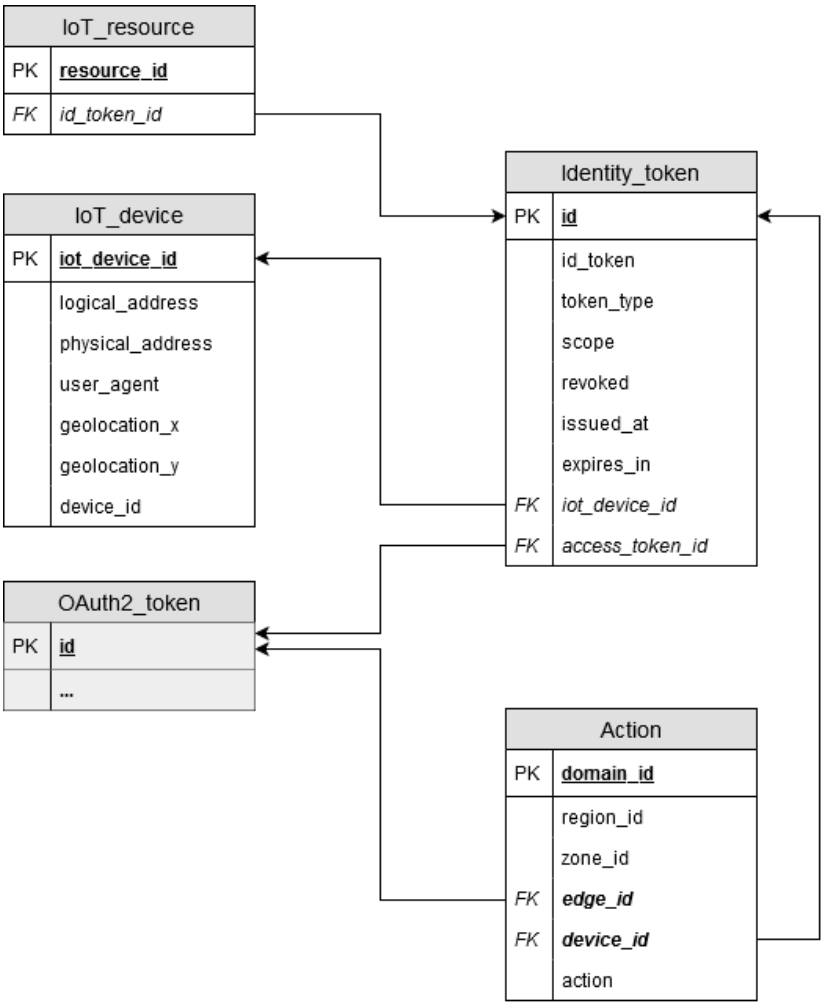

Figura 5.5: Extensión propuesta del modelo de datos del estándar de OAuth 2.0

5.5 se muestra dicha extensión realizada del modelo de datos partiendo como base de la relación entre la nueva funcionalidad y la entidad que representa el *token* de acceso de OAuth 2.0.

El modelo de datos añadido al estándar se compone de cuatro entidades que se relacionan entre sí para aportar el valor añadido requerido al modelo:

- Dispositivo (*IoT_device* en la figura 5.5): al plantear un modelo de control de acceso basado en *tokens* que a su vez se enriquece de información del contexto del dispositivo, esta entidad se encarga de representar los di-

ferentes dispositivo de recursos limitados que quieren obtener el acceso a los diferentes recursos protegidos del *cloud*. Como se puede observar, se ha incluido un conjunto de información de contexto del dispositivo como campos de la entidad pero en función del dominio de aplicación y la necesidad, dicha información podría ser ampliada sin problemas.

- *Token* de identidad (*Identity_token* en la figura 5.5): esta entidad representa el *token* de identidad mediante el cual un dispositivo IoT crea o consume diferentes recursos protegidos del *cloud* a través de los dispositivos *edge*. La delegación de autorización de los dispositivos *edge* se representa de manera gráfica a través de la relación existente entre los *tokens* de acceso de OAuth 2.0 y los *tokens* de identidad. Por otro lado, los *tokens* de identidad, como ya se ha comentado, se relacionan de igual manera con la entidad que representa los dispositivos ya que son éstos los que tienen la autorización delegada en base al *token* de identidad.

- Recurso (*IoT_resource* en la figura 5.5): el conjunto de recursos protegidos disponibles para el ecosistema IoT que utiliza este modelo de datos se representa mediante esta entidad. En ella se puede observar cómo un recurso protegido se compone de un identificador de recurso y está relacionado con un y sólo un *token* de identidad. Esta relación permite llevar a cabo el control de acceso basado en tiempo de validez que se ha expuesto en este trabajo ya que al expirar o ser revocado un *token* de identidad, el recurso seguirá estando disponible para su procesamiento por parte del entorno *cloud* pero dejará de estarlo para el dispositivo IoT que habrá obtenido otra autorización delegada distinta materializada mediante un nuevo *token* de identidad.

- Acción (*Action* en la figura 5.5): finalmente, para materializar el modelo de direccionamiento basado en eventos, se ha incluido en el modelo esta entidad. Como las órdenes o acciones indicadas por el *cloud* a los diferentes dispositivos IoT deben ser dadas a dispositivos activos, se puede observar en el modelo cómo dicha acción concreta se relaciona tanto con el *token* de acceso vigente del dispositivo *edge* como al *token* de identidad del dispositivo IoT. Además, en dicha entidad se han añadido los campos necesarios para

que el esquema de nombrado sea entendible por parte del *cloud* y éste pueda llevar a cabo las distintas acciones en el entorno simplemente realizando las búsquedas en su propio modelo.

5.1.2. Rol *edge*

Dentro del rol *edge* podemos encontrar todos aquellos dispositivos de borde ya sean *gateways*, controladores independientes o embebidos, *clusters* o micro centro de datos cuya principal función es la de ser la puerta de entrada de los dispositivos IoT embebidos en el mundo físico hacia los recursos disponibles en el *cloud*. Con el fin de cumplir dicha función, estos dispositivos tienen que ser capaces de entender tanto los protocolos de comunicación que soportan los entornos *cloud* como los protocolos de comunicación más ligeros soportados por los dispositivos IoT. Por este motivo, la funcionalidad requerida tanto para este rol como para el típico dispositivo IoT se ha implementado apoyada en el proyecto CoAPthon [173]. A dicho proyecto ha sido necesario añadirle la funcionalidad para servir de *proxy* entre CoAP y HTTP/HTTPS, la cual era una funcionalidad necesaria para la implantación de la propuesta realizada en esta tesis y que no estaba cubierta por dicho trabajo.

El desarrollo llevado a cabo del dispositivo *edge* proporciona, como ya se ha comentado, la traducción de protocolo CoAP a HTTP y viceversa, además de llevar a cabo la correcta gestión de la cabecera CoAP *ETag* la cual es utilizada como portadora del *token* de identidad del dispositivo IoT en sus accesos a los recursos *cloud*. Con el fin de gestionar los diferentes contextos de los dispositivos de recursos limitados en las diferentes validaciones, el dispositivo *edge* también cuenta con una sencilla caché. Por otro lado, se ha desarrollado también el mínimo de funcionalidad requerida de un dispositivo IoT para probar y validar los diferentes flujos ya descritos en el capítulo anterior. El código fuente tanto del rol *edge* como del dispositivo de recursos limitados se encuentra disponible en [174].

APIs definidas

Para que un dispositivo IoT pueda registrarse en el sistema e interactuar con los diferentes recursos proporcionados por el entorno *cloud*, el dispositivo *edge*

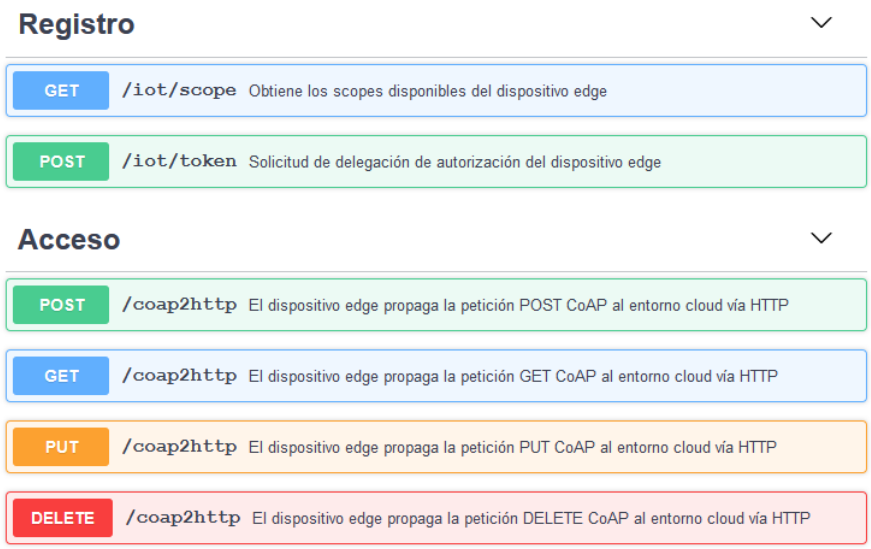

Figura 5.6: APIs definidas para el rol *edge*

proporciona una serie de APIs CoAP. Dichas APIs independizan al dispositivo de recursos limitados de los protocolos de comunicación computacionalmente más costosos dadas sus características intrínsecas y abstrae de la complejidad del protocolo OAuth 2.0 a dichos dispositivos. En la figura 5.6 se muestra la descripción *swagger* de las APIs divididas en dos subconjuntos: las APIs necesarias para llevar a cabo el registro del dispositivo IoT y las APIs necesarias para acceder a los recursos disponibles en el *cloud*.

A diferencia de las APIs de acceso a los recursos *cloud*, las APIs definidas para el flujo de registro de un dispositivo IoT no requieren de ningún tipo de cabecera CoAP ya que toda la información necesaria se puede gestionar mediante los parámetros de entrada y salida, como por ejemplo, la información de contexto del dispositivo que se está registrando. Por otro lado, las cuatro APIs de acceso si que requieren de las dos cabeceras CoAP para su correcto funcionamiento: *ETag* y *Proxy-Uri*. Por un lado, la cabecera *ETag* contiene el *token* de identidad del dispositivo IoT que luego será enviado al *cloud* a través de la cabecera HTTP *IoT-Forwarded* descrita en el rol *cloud*. Por otro lado, la cabecera *Proxy-Uri* contiene la URI del recurso concreto del *cloud* al que el dispositivo IoT quiere acceder, siendo dicha URI hacia la que el dispositivo *edge* realizará su petición HTTP

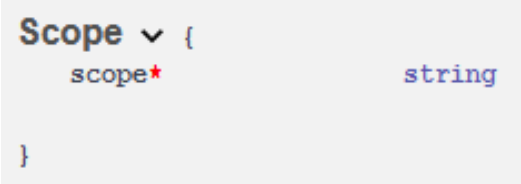

Figura 5.7: Definición del objeto JSON *Scope*

Figura 5.8: Definición del objeto JSON *Token_Request*

junto a su *token* de acceso y al *token* de identidad del dispositivo IoT.

Con respecto a cuáles son los parámetros de entrada y de salida de cada una de las APIs descritas más allá de las cabeceras CoAP, a continuación se describe dicho conjunto de parámetros concreto para cada una de ellas:

- GET */iot/scope*: la función de este servicio es que el dispositivo IoT que quiere formar parte del sistema pueda conocer los *scopes* o roles de OAuth 2.0 que tiene disponible el dispositivo *edge* sobre el que va a realizar el flujo de registro para establecer cuáles de ellos necesita dentro de sus funciones. Este método es la primera invocación obligatoria por parte del dispositivo IoT dentro del flujo de registro y podría ser un GET o un POST en función de la implementación requerida. La diferencia básica sería que en caso de ser un GET, el dispositivo *edge* obtiene la información de contexto posible del dispositivo IoT a través de la propia petición en sí, y si fuera un POST, es el dispositivo IoT quién le proporciona información extra en el cuerpo del mensaje. Una vez el dispositivo *edge* ha cacheado la información de contexto del dispositivo IoT, retorna a dicho dispositivo el objeto *Scope* mostrado en la figura 5.7 que representa la información de qué roles de OAuth 2.0 están disponibles en el dispositivo *edge* consultado.

- POST */iot/token*: dentro del flujo de registro de un dispositivo IoT en el sistema, esta sería la segunda invocación obligatoria por parte de dicho dispositivo. Esta API es realmente la encargada de apantallar la complejidad de OAuth 2.0 para los dispositivos IoT en su registro. Lo primero que se realiza en este servicio es extraer del objeto *Token_Request* mostrado en la figura 5.8 de la petición enviada por el dispositivo IoT la información del contexto de dicho dispositivo que se encuentra representada mediante el objeto *IoT_Device* (figura 5.2) dentro del campo *metadata*. Dicha información de contexto se valida para comprobar si el dispositivo IoT realizó previamente la invocación a la API */iot/scope* como se indica en el flujo de registro. Si dicha invocación fue llevada a cabo correctamente, se revisa que el *scope* o rol de OAuth 2.0 seleccionado por el dispositivo IoT está disponible en el dispositivo *edge* consultado y de ser así, se realiza la invocación al *cloud* correspondiente para continuar con el flujo de registro. Finalmente, una vez concluido el registro del dispositivo IoT, el *cloud* retornará al dispositivo *edge* el *token* de identidad y éste a su vez le devolverá dicho *token* al dispositivo IoT a través del objeto *Id_Token* mostrado en la figura 5.3.

- POST | GET | PUT | DELETE */coap2http*: este servicio se encarga de traducir la petición CoAP llevada a cabo por el dispositivo IoT a la petición HTTP correspondiente que entenderán los servicios disponibles en *cloud*. De la misma forma, cuando el dispositivo *edge* reciba la respuesta del *cloud* éste la traducirá nuevamente de HTTP a CoAP. Por tanto, la especificación de esta API sólo requiere que el cuerpo del mensaje o el de la respuesta, en función del método CoAP utilizado, esté en formato JSON y que en dicha petición realizada por parte del dispositivo IoT se hayan incluido las dos cabeceras *ETag* y *Proxy-URI* ya mencionadas.

Finalmente, una vez descritas las APIs que dan soporte al flujo de registro y de acceso de los dispositivos IoT a través de los dispositivos *edge*, es necesario poner foco en el conjunto de APIs de servicio que utilizan los propios dispositivos *edge* entre ellos para propagar las diferentes peticiones cuando el dispositivo IoT que está intentando acceder a los recursos del *cloud* está en *roaming*. En la figura 5.9 se muestra la descripción *swagger* de las APIs HTTP que sólo los dispositivos

Figura 5.9: APIs definidas para la comunicación *Edge2Edge*

edge pueden invocarse entre ellos, es decir, nunca un dispositivo IoT tendrá estas APIs disponibilizadas en CoAP ni podrá invocarlas directamente.

Desde el punto de vista de restricción en los campos de entrada y de salida, dichas APIs de servicio sólo requieren que el cuerpo del mensaje o el de la respuesta, en función del método HTTP utilizado, esté en formato JSON. Por otro lado, desde el punto de vista de cabeceras HTTP necesarias para el correcto funcionamiento de estas APIs, el dispositivo *edge* que realiza la petición debe incluir en la cabecera HTTP *IoT-Forwarded* el *token* de identidad del dispositivo IoT que está en *roaming* y en la cabecera HTTP *Referer* incluirá la URI que había sido indicada por el dispositivo IoT en la cabecera CoAP *Proxy-URI*. Por último, y en función de la exposición y consideraciones de seguridad que se quieran aplicar a estas APIs de servicio, dichas APIs pueden estar filtradas por IP, por lo que no sería necesario añadir ninguna cabecera HTTP adicional, o estar protegidas por OAuth 2.0, lo cual requeriría añadir la cabecera HTTP *Authorization* con el *token* de acceso del dispositivo *edge* que realiza la petición.

Modelo de datos y persistencia

A diferencia del modelo de datos planteado para el rol *cloud* en el cual es necesaria la persistencia del mismo en una base de datos junto al resto entidades del modelo de OAuth 2.0, para el rol *edge*, el modelo de datos se ha planteado como un conjunto de datos en caché, es decir, persistencia en memoria y no en disco, que busca optimizar al máximo posible el rendimiento del esquema de delegación de autorización planteado en este trabajo de tal forma que permita a cada dispo-

sitivo *edge* una mejor gestión escalable del gran número de dispositivos IoT que puede apantallar.

Desde el punto de vista de configuración general de la caché configurada en los dispositivos *edge*, dicha caché basa la gestión de sus entradas en el algoritmo LRU (*Least Recently Used*), es decir, en caso de necesitar eliminar alguna de las diferentes entradas disponibles en la caché por encontrarse llena, se comenzará eliminando las entradas menos recientemente utilizadas. Esto es así ya que, traducido al contexto del Internet de las cosas, puede que el dispositivo IoT se haya quedado sin batería, haya sufrido algún tipo de problema técnico al estar embebido en el mundo físico como pérdida total de cobertura, o incluso se encuentre en *roaming* durante tanto tiempo que finalmente se haya terminado registrando en el sistema de nuevo pero a través de otro dispositivo *edge*.

Con respecto al tiempo de retención de los datos en la caché, dicho parámetro está delimitado totalmente por el tiempo de vida del *token* de identidad que le confiera el servidor de autorización de OAuth 2.0, es decir, cuando un dispositivo IoT se registra en el sistema y obtiene su *token* de identidad, su contexto es el que queda almacenado en el *cloud* y cacheado en el dispositivo *edge* para optimizar el rendimiento total del sistema. Cuando dicho *token* de identidad expira, el propio dispositivo IoT vuelve a ejecutar el flujo de registro. Durante este nuevo registro puede que algún parámetro de su contexto haya cambiado de la vez anterior y por este motivo, el tiempo de retención de las entradas en la caché nunca se debe configurar con un valor superior al tiempo de vida del *token* de identidad que se defina en el dominio de aplicación concreto ya que se estaría almacenando de manera ineficiente información que ya no sería útil para el sistema.

Finalmente, respecto al tamaño concreto de cada entrada que tiene la caché teniendo en cuenta el par clave-valor en su conjunto, se trataría de un tamaño por entrada de 444 bytes, donde 134 bytes serían para la clave de la caché que contiene el *token* de identidad y 310 bytes para albergar el contexto del dispositivo IoT. Considerando que todos los campos descritos en el objeto *IoT_Device* mostrado en la figura 5.2 vienen informados o se pueden obtener del contexto a la hora de registrar el dispositivo IoT, la tabla 5.1 muestra el desglose a bajo nivel de bytes que ocupa cada campo de contexto del dispositivo IoT bajo el modelo propuesto. Este tamaño de entrada en caché implica que para el caso de un dispositivo

Tabla 5.1: Desglose en bytes de cada objeto en caché de un dispositivo *edge*

Tipo	**Dato**	**Tamaño (Bytes)**
Clave de la caché	*id_token*	134
Contexto del dispositivo (valor)	*logical_address*	4
	physical_address	18
	user_agent	256
	geolocation_x	8
	geolocation_y	8
	device_id	16

edge que esté funcionando en paralelo con 10.000 dispositivos IoT, el tamaño de memoria que le consumiría mantener en caché el contexto de dicha cantidad de dispositivos sería tan solo de 4,23 megabytes, lo que supone un uso eficiente de los recursos de cada dispositivo *edge*.

5.2. Validación y evaluación en el primer caso de uso

5.2.1. Caso de uso: Agricultura inteligente

Los dispositivos IoT ofrecen a la agricultura y ganadería una potente herramienta para obtener un mejor control de sus procesos, aumentando su eficiencia y la calidad de los mismos, y reduciendo así los riesgos. Es por ello que este primer caso de uso se centra en una aplicación de monitorización de ganado vacuno que se encarga de monitorizar en remoto dichas vacas con mecanismos de alerta. El objetivo principal es la medición de manera constante de la temperatura, la actividad y el comportamiento (movimiento de la cabeza, desplazamientos, nutrición, proximidad a otras vacas) de cada vaca individual utilizando dispositivos IoT en forma de collares de cuello inteligentes y no invasivos.

Toda la información recopilada se carga en los servidores *cloud* centrales de la solución para su almacenamiento, procesamiento y análisis; permitiendo a los agricultores iniciar diversas estrategias para mejorar el cuidado y la productividad del ganado. Por ejemplo, se pueden modelar los ciclos reproductivos de las

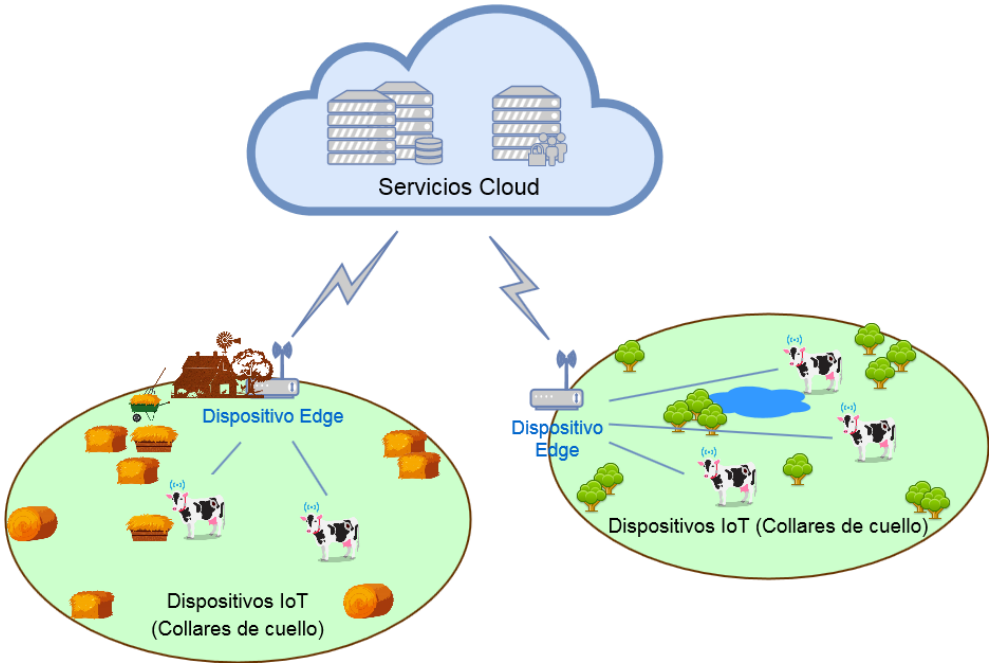

Figura 5.10: Actores implicados en el caso de uso de agricultura inteligente

vacas con el fin de predecir épocas de apareamiento o parición (para que el veterinario pueda estar preparado en el caso de que las explotaciones ganaderas estén alejadas de los núcleos poblacionales). También se pueden modelar los ciclos de producción de leche, ajustando la dieta de las vacas para que sea beneficiosa para su salud y optimizando su producción.

Para terminar, cabe destacar que los dispositivos *edge* que sirven de enlace entre los dispositivos IoT que portan cada una de las vacas y los propios servidores *cloud* centrales del sistema se encuentran desplegados por las explotaciones ganaderas de tal forma que pueden cubrir toda la superficie de la misma. De esta forma, este caso de uso cubre al completo el esquema de delegación de autorización propuesto en el capítulo 3 incluyendo incluso, el flujo de acceso a los recursos en *roaming*. La figura 5.10 muestra de manera gráfica este caso de uso concreto en el que la explotación ganadera dispone de dos zonas para las vacas: un establo y una zona de pasto.

5.2.2. Delegación de autorización para dispositivos IoT

Una vez descritos los aspectos funcionales y detalles técnicos del esquema de delegación de autorización propuesto, el siguiente paso es el de describir el proceso de validación del mismo y el análisis de rendimiento que se ha llevado a cabo. Para ello, antes de empezar, se describe a continuación las características técnicas de los diferentes dispositivos utilizados en función del rol que ocupan dentro de la arquitectura de referencia definida:

- Servicios *cloud*: tanto los servicios *cloud* como el servidor de OAuth 2.0 enriquecido con la extensión propuesta en este trabajo han sido desplegados sobre un portatil Intel(R) Core(TM) i7-6700 CPU, 3.40 GHz, con 16 GB DDR4. Como exige la especificación estándar de OAuth 2.0 y el esquema de delegación de autorización descrito en esta tesis, todos los servicios expuestos en este servidor soportan comunicación HTTPS.

- Dispositivos *edge*: estos dispositivos que se encargan de actuar como intermediarios traductores entre los dos mundos; el mundo físico del Internet de las cosas y los servicios *cloud*, se basan en Raspberry Pi 1 Model B+ con 1 GB LPDDR2 SDRAM. Además cuentan con la capacidad de disponer de módulos de batería cuya duración media es de seis meses o en su defecto, disponen de conexión a la red eléctrica, por lo que no existe preocupación en estos casos de un uso menos eficiente de la batería. Dichos dispositivos se encuentran desplegados próximos a los dispositivos IoT de recursos limitados, lo cual permite a estos últimos llevar a cabo comunicaciones eficientes y de baja latencia utilizando CoAP.

- Dispositivos IoT: el hardware utilizado en este tipo de dispositivos de recursos limitados ofrece obviamente muchas más restricciones que los dispositivos *edge* ya mencionados. Más concretamente, los dispositivos IoT utilizados se basan en un microcontrolador de 32 bits de CPU, con 128 KB de RAM, 8 MB de memoria flash como almacenamiento persistente. Además disponen de una batería que les proporciona una autonomía de unos 12 meses de media.

Usando los dispositivos descritos se ha llevado a cabo un conjunto de pruebas cuyo fin ha sido el de comprobar la viabilidad técnica del esquema propuesto y extraer un conjunto de conclusiones sobre posibles virtudes y puntos de mejora relacionados con el rendimiento del esquema de cara a su explotabilidad futura en un entorno productivo real. Aunque el objetivo principal de las pruebas se ha puesto en definir el modelo de amenazas que ayude a determinar el nivel de madurez desde el punto de vista de la seguridad del esquema de delegación de autorización propuesto, la necesidad de determinar los puntos fuertes y de mejora desde el punto de vista de rendimiento del sistema se hace necesario en un contexto tan exigente como es el Internet de las cosas. En las siguientes subsecciones se abordan las pruebas realizadas tanto para la parte de gestión de identidades y accesos así como para el direccionamiento y nombrado.

Gestión de identidades y accesos en IoT

El conjunto de experimentos que se ha llevado a cabo para validar los flujos de registro, de acceso a recursos y de acceso en *roaming* así como sus propiedades requeridas, han sido diseñados considerándose diferentes escenarios y configuraciones. Los escenarios planteados han buscado cubrir los diferentes enfoques de pruebas funcionales, de integración y de rendimiento. Para el primer conjunto de pruebas que cubren el aspecto funcional y de integración, se ha utilizado un dispositivo *edge* y dos dispositivos IoT. Por otro lado, en el segundo conjunto de pruebas se ha llevado el escenario hasta un despliegue de 100 dispositivos *edge* y 1.000 dispositivos IoT con el fin de cubrir las pruebas de rendimiento del sistema. En este último conjunto, las pruebas se han realizado en un entorno donde las comunicaciones de red son ideales, es decir, no existe ningún tipo de interferencias o interrupciones en las comunicaciones que puedan suponer una variable generadora de entropía en los resultados de tiempos medios obtenidos.

Durante las pruebas se ha comprobado que los servicios *cloud* son capaces de autenticar y autorizar a los dispositivos IoT a través de los dispositivos *edge* de una manera completamente transparentes sin necesidad de conocer los detalles de implementación y protocolos de comunicación utilizados por los dispositivos IoT. Esto es así ya que los entorno *cloud*, al no tener conexión directa nada más que

Tabla 5.2: Análisis de rendimiento del esquema definido: latencia

Flujo analizado	**Tiempo medio**	**Desviación estándar**
Registro	1 s	200 ms
Acceso a recursos	800 ms	180 ms
Acceso en *roaming*	1,25 s	195 ms

con los dispositivos *edge*, están completamente abstraídos de cualquier cambio en la topología de red, características de las comunicaciones o cualquier otro cambio que suceda en los dispositivos IoT, lo cual permite que el mismo enfoque del rol *cloud* planteado pueda ser utilizado en diferentes dominios de aplicación sin necesidad de ser modificado.

Por otro lado, en las pruebas de rendimiento se han llevado a cabo 1.000 ejecuciones de manera paralela para cada uno de los diferentes flujos, siendo los tiempos medios de respuesta y la desviación estándar de lujos los que se muestran en la tabla 5.2. Al comprobar durante las diferentes ejecuciones que el registro de los dispositivos IoT se lleva a cabo de manera automatizada y sin intervención humana, se demuestra la premisa inicial del esquema en la cual se buscaba la garantía de los niveles de escalabilidad deseados acordes al contexto del Internet de las cosas.

Los tiempos medios ayudan a tener una visión global del comportamiento del sistema, sin embargo, se hace necesario entender también el desglose de ese tiempo para poder identificar puntos de mejoras en el esquema. La tabla 5.3 muestra el porcentaje del tiempo medio en cada flujo indicado en la tabla 5.2 que consume cada una de las partes que conforman el esquema de delegación de autorización, es decir, los servicios *cloud* y servidor de OAuth 2.0, los dispositivos *edge* y las propias comunicaciones. Hay que destacar de estos resultados, que tanto los tiempos medios de respuesta de los servicios *cloud* así como el de los dispositivos *edge* podrían llegar a optimizarse en cierta medida, pero por otro lado, en función de las condiciones y la complejidad de las topologías de red que finalmente conforme el entramado de los dispositivos IoT de recursos limitados, el porcentaje del tiempo invertido en las comunicaciones podría verse aumentado, repercutiendo por tanto, en el tiempo medio de los diferentes flujos.

Con respecto al uso de la memoria y el *overhead* en las comunicaciones que

Tabla 5.3: Desglose del tiempo medio de respuesta de cada rol del esquema

Flujo analizado	**Servicios *cloud***	**Dispositivos *edge***	**Comunicaciones**
Registro	30 %	25 %	45 %
Acceso a recursos	37,5 %	12,5 %	50 %
Acceso en *roaming*	24 %	20 %	56 %

sufre el dispositivo IoT al hacer uso del esquema de delegación de autorización diseñado, se observa la posibilidad de aplicar dos casos de uso en función de las limitaciones intrínsecas de los propios dispositivos sin conllevar una penalización en el uso eficiente de sus limitados recursos. En primer lugar, el caso más eficiente que cubre las restricciones más exigentes de un dispositivo IoT se basa en la custodia en memoria del *token* de identidad de 134 bytes, el cual es la única información que éste añade en cada petición para identificarse a través de los dispositivos *edge* cuando requiere consumir recursos disponibles en el *cloud*. Este caso se considera el más restrictivo ya que el contexto que recaba el dispositivo *edge* en el flujo de registro se reduce sólo a la información de contexto que puede obtener de la propia petición, la cual puede constar en el peor de los casos, sólo de la dirección lógica del dispositivo IoT. Por otro lado, en los casos en los que no exista tanto nivel de restricción por parte de los dispositivos IoT y si la implementación lo permite, se puede añadir al *overhead* en las comunicaciones 310 bytes en las peticiones para hacer partícipe al dispositivo *edge* del contexto enriquecido que tiene dicho dispositivo IoT.

Finalmente, como cada dispositivo IoT almacena y envía a través de la red sólo los *tokens* de identidad y a lo sumo, su información de contexto, no se identifica ningún tipo de anomalía o eventualidad en el consumo de energía, ya que en el esquema planteado, el dispositivo IoT no tiene que realizar ningún tipo de operación criptográfica compleja o cálculo computacionalmente costoso con los *tokens* de identidad, sino simplemente, ser portador de los mismos. El único punto de mejora u optimización identificado hasta este punto estaría relacionado con la ampliación del tiempo de vida de los *tokens* de identidad que se derivan a los dispositivos IoT. Esto reduce el número de veces que un dispositivo IoT ejecuta el flujo de registro para obtener un nuevo *token* de identidad cuando el suyo expira, mejorando así, el correspondiente consumo de recursos que conlleva la actualiza-

ción de su *token* de identidad. Cabe destacar, que para no corromper el esquema propuesto desde el punto de vista de seguridad, habrá que establecer el tiempo de vida más restrictivo posible en función del dominio de aplicación concreto en el que se aplique este esquema de delegación de autorización.

Direccionamiento y nombrado en IoT

Para llevar a cabo la validación del esquema de direccionamiento y nombrado de IoT propuesto se ha llevado a cabo el mismo conjunto de experimentos que para el caso de los flujos del esquema de delegación de autorización. Por un lado, para cubrir el escenario de validación de las pruebas funcionales y de integración se ha utilizado un dispositivo *edge* y dos dispositivos IoT, y para cubrir el escenario de las pruebas de rendimiento, se ha llevado a cabo el despliegue de 100 dispositivos *edge* y 1.000 dispositivos IoT en las mismas condiciones ideales de las comunicaciones de red, es decir, sin existir ningún tipo de interferencias o interrupciones en las comunicaciones que puedan suponer una variable generadora de entropía en los resultados de tiempos medios obtenidos.

Desde el punto de vista de rendimiento del flujo de actuación, al ser un flujo idéntico desde el punto de vista funcional al flujo de acceso a recursos y como se ha replicado el mismo escenario de pruebas en las mismas condiciones, los tiempos medios de respuesta y la desviación estándar de dicho flujo coinciden con los indicados para el flujo de acceso a los recursos mostrado en la tabla 5.2 teniendo incluso, la misma distribución de porcentaje de tiempo de ejecución dentro del flujo por cada una de las partes involucradas como se muestra en la tabla 5.3. Estos resultados nos permiten extrapolar al flujo de actuación las mismas conclusiones ya descritas de los flujos englobados dentro del esquema de delegación de autorización analizado en la sección anterior.

Más allá de los resultados ya comentados desde el punto de vista de rendimiento, se observan tres aspectos sumamente interesantes que refuerzan la robustez de dicho modelo de direccionamiento y nombrado propuesto:

- Garantía de entrega de mensajes: las soluciones que envían de manera proactiva los mensajes desde los servicios *cloud* hasta los dispositivos IoT embebidos en el mundo físico ignoran si los mensajes han llegado finalmente

a su destino o no debido, por ejemplo, a problemas de red o de traducción de protocolos de comunicación, o incluso, que el propio dispositivo IoT esté inoperativo o cualquiera de los intermediarios entre el *cloud* y dicho dispositivo y no exista una ruta alternativa de entrega de mensajes. Por el contrario, el esquema de direccionamiento propuesto permite la capacidad a los servicios *cloud* de conocer en todo momento qué dispositivo IoT está activo y cuál no. Esto es así ya que si un dispositivo IoT tiene un *token* de identidad válido, si se observa que durante un largo periodo de tiempo éste no accede a consumir recursos disponibilizados en el *cloud* ni obtiene las diferentes acciones indicadas por el *cloud* que debe realizar para interactuar en el mundo físico, los propios servicios *cloud* pueden revocar dicho *token* debido a la inactividad observada, minimizando así el riesgo de fuga de información y pudiendo incluso ejecutar medidas para entender por qué los dispositivos no están funcionando, es decir, el *cloud* tiene constancia en todo momento de qué dispositivo está activo y ha obtenido las acciones que debe llevar a cabo gracias al propio uso que realiza de su *token* de identidad.

- Gestión de caché jerárquica: gracias a la organización jerárquica existente entre dispositivos *edge* y dispositivos IoT tanto desde el punto de vista de arquitectura como del esquema de nombrado propuesto, el uso de una caché jerárquica encaja perfectamente en este diseño. Dicha caché permitiría que, por ejemplo, cuando los servicios *cloud* requieren que una acción se aplique a todos los dispositivos IoT registrados a través de un dispositivo *edge*, dicha acción podría ser cacheada por dicho dispositivo *edge* de tal forma que permitiría liberar recursos tanto de comunicación como de cómputo al *cloud* y lo más importante, mejoraría el tiempo de respuesta del flujo de actuación para que los dispositivos IoT puedan obtener de una manera mucho más en tiempo real cómo deben interactuar con el mundo físico.

- Gestión de tolerancia a fallos o *failover*: la organización jerárquica ya introducida permite una gestión de tolerancia a fallos en función de los diferentes niveles de la jerarquía. Si las regiones o las zonas son consideradas como áreas de despliegue de servicios *cloud*, cuando exista una pérdida del servicio en cualquiera de ellas, los dispositivos *edge* podrían conectarse a

otra y seguir trabajando si dichos servicios *cloud* se hubieran desplegado cubriendo varias de estas áreas de disponibilidad. De la misma forma, si se despliega de manera conjunta tanto el esquema de direccionamiento y nombrado como la solución de acceso en *roaming*, si un dispositivo *edge* falla en su interfaz de comunicación CoAP, los dispositivos IoT tendrían la capacidad de utilizar otro dispositivo *edge* de intermediario o incluso, de registrarse de nuevo en el sistema a través de otro dispositivo *edge*.

5.2.3. Análisis de seguridad

Los experimentos llevados a cabo con el fin de demostrar las propiedades desde el punto de vista de seguridad del esquema planteado se basan tanto en el modelo de amenazas STRIDE [175] definido por Microsoft como en el conjunto de árboles de amenazas habituales a revisar por cada uno de los diferentes elementos que componen este sistema. En dichos experimentos de seguridad se han intentado materializar sobre el esquema de delegación de autorización propuesto, todas las posibles amenazas utilizando diferentes vectores y categorías de ataques [176].

En las tablas 5.4 y 5.5 se resume el modelo de amenazas formal realizado para abordar los diferentes experimentos sobre el esquema de delegación de autorización propuesto donde, la primera columna representa la amenaza o riesgo analizado siguiendo el acrónimo STRIDE, la segunda es el modo elegido para materializar la amenaza, la tercera es el ataque concreto y la cuarta, muestra si finalmente el ataque tendría éxito o no sobre el sistema. En el caso de que exista un Sí en la columna de éxito, significa que el atacante es capaz de materializar la amenaza y obtener los resultados deseados usando el modo concreto de ataque correspondiente en la fila. Por otro lado, un No en dicha columna significa que un atacante no tendría éxito al materializar un ataque de esa manera. Finalmente, la quinta columna cualifica la dificultad de materializar dicho ataque en un entorno productivo, siendo los niveles de dificultad: B (bajo), M (medio), y A (alto).

Con el fin de detallar los puntos clave identificados a lo largo de los diferentes experimentos de seguridad realizados, a continuación se lleva a cabo un análisis en profundidad de dichos puntos añadiendo también, las consideraciones de se-

Tabla 5.4: Análisis de amenazas del esquema basado en el modelo STRIDE (I)

Amenaza	Modo	Ataque	Éxito?	Dif.
Suplantación	Robo de credenciales	En tránsito	No	-
		Problemas de federación	No	-
		Gestión de cambios	No	-
		Almacenamiento	Sí	B
	Autenticación (AuthN.)	Login local	No	-
		Acceso privilegiado	No	-
		Suplantación remota	Sí	B
	AuthN. insuficiente	Credenciales Null	No	-
		Creds. de invitado	No	-
		Creds. predecibles	Sí	M
		Creds. defecto fábrica	No	-
		AuthN. por *downgrade*	No	-
	Refresco de AuthN.	Conocimiento de AuthN.	No	-
		AuthN. encadenada	No	-
		Fuga de información	No	-
	Usuario anónimo	-	No	-
	Otros	-	No	-
Manipulación	Canal	No integridad en el canal	No	-
		Integridad de canal débil	No	-
		MitM	Sí	A
	Mensaje	No integridad de mensajes	Sí	A
		Gestión débil de clave	No	-
	Basada en tiempo	Replay	Sí	M
		Reflexión	No	-
		Colisiones	No	-
	Inyecciones	-	No	-
	Otros	-	No	-

guridad necesarias sobre el esquema de delegación de autorización propuesto en esta tesis:

- Suplantación de dispositivos IoT: esta amenaza podría ser materializada mediante el robo de los credenciales (*token* de identidad) del dispositivo IoT, exponiendo una interfaz de autenticación falsa a dichos dispositivos para el flujo de acceso a los recursos, explotando un insuficiente control de autenticación a la hora de emitir el *token* de identidad o atacando el proceso de actualización o refresco de la autenticación.

  En primer lugar, al estar las comunicaciones CoAP protegidas por DTLS entre el dispositivo IoT y el dispositivo *edge*, y las comunicaciones REST

Tabla 5.5: Análisis de amenazas del esquema basado en el modelo STRIDE (II)

Amenaza	**Modo**	**Ataque**	**Éxito?**	**Dif.**
Repudio	DoS en *logs*	-	Sí	B
	Manipulación *logs*	-	Sí	B
	Transferencia	Sin *logs*	No	-
		Insuficientes *logs*	No	-
		Logs dispersos	No	-
		Logs no sincronizados	No	-
		Logs no capturan AuthN.	No	-
	Otros	-	No	-
Deneg. de servicio	Mensaje corrupto	Sin integridad	No	-
		Integridad débil	No	-
	Pre-play	-	Sí	B
	Inhabilitación canal	Consumo de recursos hw.	Sí	A
		Consumo de recursos sw.	Sí	A
		Incapacitar endpoints	Sí	B
	Otros	-	No	-

por HTTPS entre el dispositivo *edge* y los servicios *cloud*, la única forma de materializar el robo u obtención de los credenciales del dispositivo IoT al no estar basada la autenticación en una solución federada, sería atacando directamente el almacenamiento en memoria del propio dispositivo IoT. El *token* de identidad se almacena en la memoria del dispositivo IoT durante todo el tiempo de vigencia del mismo y podría ser extraído con facilidad al no existir ningún tipo de control de seguridad a nivel hardware o de memoria en dicha clase de dispositivos.

En segundo lugar, sólo la suplantación remota ofreciendo la misma interfaz que un dispositivo *edge* podría garantizar la obtención en tránsito del *token* de identidad de un dispositivo IoT. Este ataque es relativamente sencillo de materializar debido a las características del diseño de la solución, y aunque un dispositivo IoT se configure por defecto para usar el mismo dispositivo *edge* que lo registró en el sistema, dicho dispositivo *edge* podría sufrir una denegación de servicio deliberada para forzar la activación del flujo de acceso en *roaming* y por tanto, ser este segundo dispositivo *edge* el dispositivo malicioso.

Por otro lado, si la generación del *token* de identidad se vincula fuertemente al contexto del dispositivo IoT que se registra y no se lleva a cabo su emisión

de una manera criptográficamente segura que evite los ataques de colisión, un atacante podría intentar predecir dicho *token* de identidad tratando de suplantar a un dispositivo IoT legítimo utilizando el mismo contexto que el dispositivo IoT legítimo haya utilizado para registrarse en el sistema.
Y en cuarto lugar, referido al ataque sobre el proceso de actualización o refresco de la autenticación, dicho ataque no sería posible sobre el esquema de delegación de autorización diseñado ya que en él se fuerza siempre la autenticación base y no existe forma de refrescar o actualizar un *token* de identidad ya emitido.
Todos estos riesgos, aunque parezca bastante sencillo que se materialicen, se mitigan por el propio diseño de la solución. Esto es así ya que si un atacante intenta suplantar la identidad de uno o varios dispositivos IoT robando sus *tokens* de identidad o suplantando su información de contexto, este comportamiento anómalo puede ser detectado por el servidor de OAuth 2.0 gracias a la existencia de la relación directa entre el *token* de identidad y las acciones permitidas por un dispositivo IoT sobre los servicios *cloud*. Por este motivo, si se detecta un ataque de suplantación en varios dispositivos IoT y el propio sistema determina que dicho ataque representa o se categoriza con un riesgo alto para el mismo, el servidor de OAuth 2.0 puede revocar el *token* de acceso del dispositivo *edge* para inhabilitar los diferentes *tokens* de identidad que se están utilizando de manera indebida. La organización jerárquica de *tokens* que plantea el esquema de delegación de autorización propuesto funciona a nivel de revocación de la misma forma que una infraestructura de clave pública cuando revoca una autoridad de certificación intermedia, es decir, si se revoca dicha autoridad de certificación intermedia la cual casaría con un *token* de acceso en este modelo, todos los certificados emitidos por ella quedarían automáticamente revocados, lo que se traduce en esta solución como todos los *tokens* de identidad ligados al *token* de acceso revocado.

- Manipulación de la integridad en el flujo de acceso a los recursos: para poder materializar este riesgo, un atacante puede intentar manipular tanto el canal como los mensajes en sí (tanto peticiones como respuestas) que ten-

gan insuficientes medidas para la integridad. Por un lado, con el fin de proteger el canal en sí, el uso de los protocolos como HTTPS y DTLS permiten cubrir los requisitos de integridad del propio canal de comunicación. Sin embargo, aunque dichos protocolos criptográficos estén debidamente configurados, se podría llevar a cabo un ataque de MitM (*Man in the Middle*) con el fin de comprometer la integridad a nivel de los mensajes intercambiados dentro del propio flujo de acceso a los datos. Dicho flujo, a nivel de diseño, no incluye protecciones específicas de ningún tipo para asegurar la integridad a nivel de mensaje ya que lo delega en la propia protección del canal. Por tanto, un atacante tendría la capacidad de modificar los mensajes enviados para obtener beneficio de ellos, aunque sería bastante difícil llegar a encontrar una modificación concreta dentro de la API RESTful diseñada que pudiera producir un beneficio directo para dicho atacante. Por otro lado, otros ataques de tipo inyección o basados en tiempo que se apoyen en técnicas de reflexión o colisión pueden ser evitados si se lleva a cabo una correcta y meticulosa implementación del flujo de acceso con los debidos controles de autenticación, validación de parámetros de entrada, y control de flujo de peticiones acorde a las capacidades limitadas de los propios dispositivos IoT. Finalmente, los ataques de *replay* son posibles dentro del flujo diseñado pero su impacto en el sistema es mínimo gracias a la característica del tiempo de vida de los propios *tokens* de identidad, la cual se debe configurar de manera segura acorde al dominio de aplicación concreto.

- Repudio de los datos almacenados (*logs* en los dispositivos *edge* y en el servidor de OAuth 2.0): la existencia de vinculación entre los *tokens* de identidad y las acciones concretas ejecutadas en los servicios *cloud* permite conocer en todo momento qué dispositivo IoT está accediendo o no a los diferentes servicios *cloud*. Si un dispositivo IoT se ve comprometido y está llevando a cabo diferentes acciones de índole sospechosa, el propio servidor de OAuth 2.0 puede saber qué dispositivo está teniendo ese comportamiento anómalo. Este gobierno del dato de bajo nivel permite tener el mejor sistema de detección de intrusiones ya que permite el poder revocar el acceso a un *token* de identidad concreto, llegando incluso a bloquear a

dicho dispositivo en futuros registros al sistema. Los únicos tipos de ataque que podrían tener éxito cuando se intenta materializar este riesgo sería la denegación de servicio o la manipulación de los *logs* que son almacenados en los propios dispositivos *edge*, los cuales tienen una protección débil ante una generación excesiva de *logs* dada su limitada capacidad de persistencia y la propia manipulación física por parte de un potencial atacante de dicha información almacenada. Sin embargo, estos riesgo puede ser mitigado por diseño ya que los *logs* más significativos y relevantes se almacenan siempre en el servidor de OAuth 2.0 al ser éste quién se encarga de la emisión de los diferentes *tokens* y de su validación posterior a la hora de llevarse a cabo el acceso a los recursos por parte de los diferentes dispositivos.

- Fuga de información de los servicios *cloud*: el gobierno del dato llevado a cabo por el modelo de delegación de autorización propuesto evita el propio minado de datos a través de cualquier dispositivo IoT comprometido. La necesidad de requerir un *token* de identidad en cada acceso a los servicios *cloud* ofrecidos y que los propios datos disponibles para cada dispositivo IoT estén vinculados al dicho *token*, restringe la superficie de exposición de los datos disponibles para un dispositivo IoT al mínimo necesario para su funcionamiento dentro del dominio de aplicación concreto en el que esté desplegado. Además, el tiempo de vida del *token* de identidad limita aún más el conjunto de datos a los que puede acceder un dispositivo IoT a lo largo del tiempo. De esta manera, al tener un dispositivo IoT capacidades limitadas de cómputo y almacenamiento, gracias a las restricciones de acceso ligadas a su *token* de identidad, dicho dispositivo sólo tendrá acceso a la cantidad de datos que sea capaz de procesar en tiempo y forma al estar limitado por diseño el acceso a la información. Esta restricción de diseño permite evitar tanto errores de programación en el software desplegado en los dispositivos IoT que puedan provocar la extenuación de los dispositivos IoT como ataques de suplantación de dichos dispositivos para llevar a cabo el minado de datos del sistema.

- Denegación de servicio en los flujos propuestos: un atacante puede usar diferentes formas de lograr que a través de los flujos definidos en el esquema

de delegación de autorización propuesto, un dispositivo *edge* no esté disponible para que los dispositivos IoT puedan interactuar con los servicios *cloud*. Este riesgo puede ser materializado, por ejemplo, con un ataque de *pre-play*, es decir, cuando un dispositivo IoT controlado por un atacante inicia una conexión con el dispositivo *edge*, dicha petición es replicada una gran cantidad de veces con el fin de extenuar los recursos de los dispositivos *edge* y agotar así el número de sesiones concurrentes disponibles que puede soportar para evitar que otros dispositivos IoT legítimos puedan registrarse o usar el sistema. También se puede materializar dicho riesgo incapacitando el propio canal de comunicación consumiendo todo el ancho de banda o incluso, agotando la propia memoria disponible de los dispositivos *edge*. Estos dos últimos ataques son muy difíciles de materializar sobre el esquema propuesto ya que los flujos diseñados son muy ligeros en términos de consumo de ancho de banda u *overhead* de las comunicaciones, y en el consumo de memoria por parte de los dispositivos *edge*. Sería mucho más sencillo incapacitar los dispositivos *edge* dentro del modelo planteado si el propio atacante pudiera tener acceso físico a dichos dispositivos para llevar a cabo su manipulación.

Por tanto, estos riesgos se mitigan parcialmente, como ya se ha comentado, en primera instancia gracias a la definición ligera de los flujos diseñados desde el punto de vista de consumo de recursos, y en segunda instancia, a la capacidad que tienen los dispositivos IoT de acceder a los recursos *cloud* en *roaming* cuando sea necesario, lo que le confiere al sistema una gran tolerancia a fallos de disponibilidad.

- Elevación de privilegios en los servicios *cloud*: el conjunto de privilegios al que puede acceder un dispositivo IoT dentro del esquema de delegación de autorización propuesto no es responsabilidad suya sino del dispositivo *edge* a través del cual se registra en el sistema. Esto es así ya que dicho dispositivo de borde es el encargado de delegarle uno o varios de los *scopes* o roles de OAuth 2.0 que dispone durante el flujo de registro. Dado que el propio *token* de acceso del dispositivo *edge* se emite asociado a un conjunto de roles de OAuth 2.0 concretos, un dispositivo IoT nunca podrá adquirir

otro rol distinto más allá de los disponibles en el *token* de acceso ya que su *token* de identidad se emitirá en base a dicho *token* de acceso y por tanto, la elevación de privilegios por parte de los dispositivos IoT en los servicios *cloud* está limitada por diseño.

5.3. Validación y evaluación en el segundo caso de uso

5.3.1. Caso de uso: Carreteras inteligentes

Dentro de los diferentes campos de interés que abarca el *crowdsensing* hay uno que destaca por encima de todos desde el punto de vista de los diferentes retos que plantea en múltiples aspectos: la monitorización y gestión del tráfico en tiempo real. Esta monitorización permite la capacidad de identificar anomalías en las diferentes vías que pueden provocarse debido a irregularidades en el pavimento, atascos, obras o accidentes, para poder tomar las decisiones oportunas en cada momento con el fin de minimizar la incidencia solucionando el problema o proponiendo rutas alternativas que permitan la descongestión del tráfico.

Por ello, este caso de uso evita los dos enfoques principales que se están tomando a día de hoy para solventar este problema. Por un lado, el de dotar a los vehículos inteligentes de una gran cantidad de sensores más allá de los necesarios para cubrir la seguridad del propio vehículos como pueden ser, los necesarios para aportar cierta retroalimentación al dominio de aplicación de las carreteras inteligentes, incrementando así la superficie de exposición de los propios vehículos. Y por otro, el del uso del paradigma del *mobile crowdsensing* mediante el cual se usa los sensores del propio dispositivo móvil del usuario para recabar la información provocando los ya conocidos problemas de privacidad debido a la necesidad del reparto de incentivos.

Por este motivo, el modelo definido en el capítulo 4 plantea la independencia de los diferentes dominios de aplicación (vehículos inteligentes y carreteras inteligentes) con el fin de buscar una mejor eficiencia de cada uno de los dominios por separado de tal forma que beneficien al bien común, apoyándose para ello en

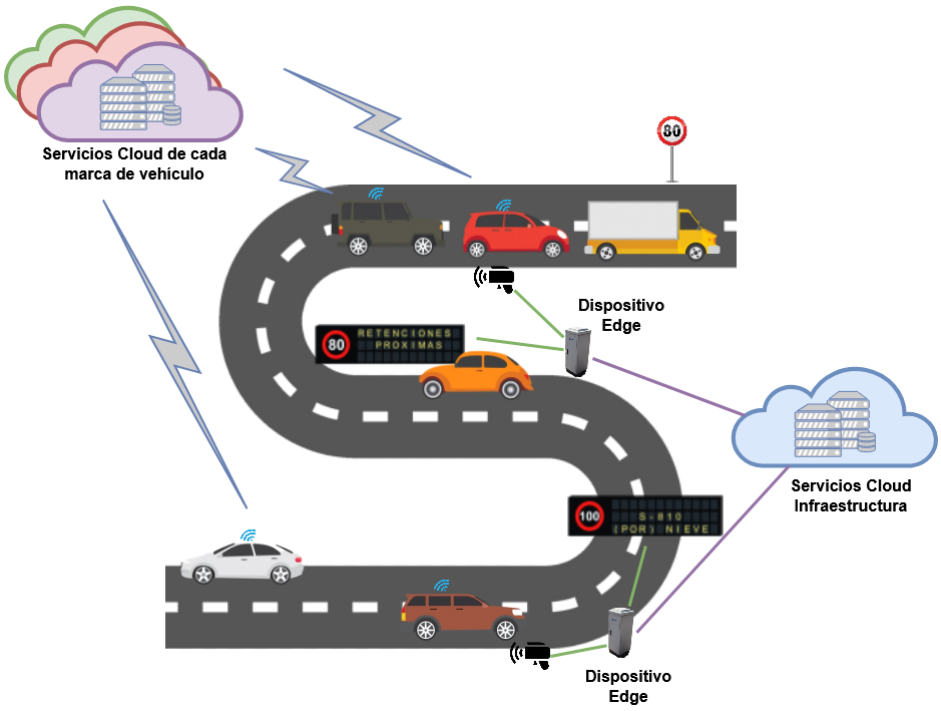

Figura 5.11: Actores implicados en el caso de uso de las carreteras inteligentes

la computación *edge* y mejorando así la toma de decisiones en tiempo real sin comprometer la privacidad del usuario ni incrementar la superficie de exposición de los vehículos inteligentes. La figura 5.11 muestra de manera gráfica, y tomando como base la arquitectura de referencia *crowdsensing* del capítulo 4, este caso de uso concreto.

5.3.2. *Crowdsensing* influenciado en entornos multidominio

Las necesidades de respuesta en tiempo real derivadas del caso de uso descrito plantea una serie de particularidades que deben ser añadidas al esquema de delegación de autorización con el fin de abordar estas nuevas características. La más importante de ellas es que en este caso los dispositivos *edge* deben ser capaces de gestionar de forma independiente los dispositivos IoT bajo su control con el fin de evitar las latencias de comunicación y de cómputo que pueda añadir el procesamiento *cloud*, siendo los dispositivos IoT, las APIs de entrada y salida de cada

dominio de aplicación que utilizan el mundo físico como medio de comunicación. En este entorno multidominio se identifican dos comunicaciones claramente diferenciadas entre ambos dominios:

- En primer lugar, los vehículos, al circular por los diferentes segmentos de la carretera, son detectados y contabilizados por la carretera inteligente independientemente de las condiciones climatológicas, de visibilidad o incluso tecnológicas de los propios vehículos. Esto se debe a la detección y reconocimiento de vehículos llevada a cabo en base a la propia huella térmica de los mismos con el uso de los sensores adecuados [177–179]. Este tipo de sensores proporcionan a la carretera la capacidad de identificar los diferentes vehículos por el propio calor de sus motores sin verse influenciados por la pérdida de visibilidad nocturna, la niebla u otras condiciones climatológicas que entorpecen claramente el uso de la detección de vehículos en base a imágenes manteniendo un enfoque respetuoso con la privacidad al no recabar identificadores como matrículas o rostros de personas en imágenes.

- Por otro lado, la carretera inteligente influencia a los diferentes vehículos que transitan por ella a través de las diferentes señales de tráfico de las que dispone para aportarles las mejores recomendaciones posibles independientemente de si el conductor es un ser humano o es un sistema autónomo. En este sentido, los vehículos autónomos están dotados de la tecnología necesaria para la lectura e interpretación en tiempo real de las señales de tráfico que los propios conductores ya han aprendido a interpretar con la experiencia [180–183]. Esto es una medio de comunicación permite, por un lado, que los vehículos inteligentes y sus servicios *cloud* sólo tienen que esforzarse en mejorar la capacidad de interpretar su entorno y adaptar su modo de conducción al contexto en tiempo real, y por otro, la carretera inteligente sólo tiene que utilizar los medios de los que dispone para llevar a cabo la obtención de información del entorno observado y sugerir las mejores recomendaciones a través de las diferentes señales de tráfico.

En último lugar, y fuera de las particularidades del mundo físico, existe la comunicación directa entre los dispositivos *edge* de los vehículos y los de la carretera inteligente con el fin de que los vehículos compartan información de telemetría

con la carretera inteligente para que ésta pueda realizar unas recomendaciones más precisas. Para este caso concreto se ha considerado la velocidad del vehículo como único dato a compartir entre ambos ya que permite a la carretera inteligente identificar la tasa de servicio de la propia carretera en tiempo real. La obtención de dicho dato por parte de un vehículo puede proceder de sus propios sensores en el caso de estar dotado de dicha tecnología o ser calculado a través de una aplicación móvil en el dispositivo del usuario. Dicha aplicación móvil requiere el permiso de geolocalización al instalarse en el dispositivo pero sólo lo utiliza para calcular la velocidad del vehículo, siendo éste, el único dato que remitirá dicho dispositivo móvil al dispositivo *edge* de la carretera inteligente. Esta comunicación entre dispositivos *edge* permite que la comunicación sea anónima como si de un sensor más se tratase ya que la geolocalización de dónde se ha tomado la muestra de la velocidad del vehículo se infiere del dispositivo *edge* de la carretera inteligente que la ha recibido. Finalmente, al no compartir ningún tipo de información sensible entre ambos dominios, el incentivo real de compartir dicho dato entre los vehículos y la carretera inteligente permite a los conductores de cualquier vehículo, autónomo o no, disponer de las recomendaciones de la carretera inteligente sin coste adicional incluso a través de la propìa aplicación móvil, la cual es considerada como un actuador más por parte del dispositivo *edge* de la carretera inteligente y por eso le remite las diferentes recomendaciones.

A partir de estas premisas sobre los dispositivos IoT y la comunicación entre los dispositivos *edge*, en esta sección se trata el conjunto de modificaciones necesarias para dotar a los dispositivos *edge* de dicha funcionalidad. Además, también se añade como estos dispositivos notifican a los entornos *cloud* de las distintas acciones de control que realizan sobre el mundo físico de manera autónoma para poder mantener la consistencia del sistema.

Rol *edge*

La necesidad de dotar al sistema o dominio de aplicación concreto de una respuesta en tiempo real conlleva una delegación de responsabilidades y funciones por parte del *cloud* a los dispositivos *edge*, como por ejemplo, la de tomar ciertas decisiones de manera autónoma para trasladarlas a los dispositivos IoT desplega-

dos en el mundo físico con una menor latencia que si lo hiciera el propio *cloud*. Las nuevas funciones adquiridas por parte de dichos dispositivos *edge* son las siguientes:

1. Para poder modificar el estado del mundo físico a través de las diferentes señales de tráfico desplegadas en él, los dispositivos *edge* deben tener disponible el conjunto de acciones permitidas que pueden realizar sobre dichas señales que tenga cada uno registradas bajo su responsabilidad.

2. Otra funcionalidad necesaria para un dispositivo *edge* es la de disponer de la información referida a las propias características de la vía en la que está desplegado, es decir, número de carriles, velocidad máxima legislada para el tipo de vía, e incluso, el porcentaje o factor de utilización permitido antes de considerar que la vía está comenzando a colapsar y hay que empezar a aplicar cambios para mitigar dicha saturación.

3. El recepción de la telemetría de la velocidad obtenida de los vehículos por parte de un dispositivo *edge* de la carretera inteligente permite disponer en tiempo real de la comparación de la velocidad máxima legislada para el tipo de vía (tasa de servicio teórica de la carretera) y la velocidad real de los vehículos (tasa de servicio actual) para poder mejorar la precisión de sus recomendaciones.

4. Finalmente, una vez el dispositivo *edge* dispone tanto de las diferentes acciones que puede realizar sobre las señales, así como de la información de la propia vía, debe ser capaz de obtener también la información del estado de saturación de al menos los dispositivos *edge* anterior y posterior a su propio tramo para poder tomar las mejores decisiones en tiempo real. Esta información debe ser facilitada por el entorno *cloud*, el cual dispone de la visión global del sistema y además, le debe proporcionar si tiene o no autorización para controlar por su cuenta el entorno que controla.

Referido al primer punto de acciones permitidas con las que puede jugar el dispositivo *edge* sobre las señales de tráfico, dicho dispositivo no expone ningún tipo de API o funcionalidad hacia los dispositivos IoT con el fin de no interferir

Tabla 5.6: Análisis comparativo de rendimiento del flujo de actuación para ambos casos de uso

Caso de uso analizado	**Tiempo medio**	**Desviación estándar**
Agricultura inteligente	800 ms	180 ms
Carreteras inteligentes	500 ms	113 ms

Tabla 5.7: Desglose en bytes de cada objeto en caché para el caso de uso de *crowdsensing*

Tipo	**Dato**	**Tamaño (Bytes)**
Clave	*id_token*	134
Valor	*logical_address*	4

en el funcionamiento ya descrito del esquema de delegación de autorización para el direccionamiento y nombrado. Teniendo de base el flujo de actuación definido anteriormente, el dispositivo *edge* además de disponer del conjunto de acciones permitidas, debe añadir a la información de contexto que dispone en su caché, la información mostrada en la tabla 5.7 que relaciona la dirección lógica del dispositivo IoT con su propio *token* de identidad. De esta forma, el dispositivo *edge* puede interceptar la petición del dispositivo IoT al *cloud* y ser él el encargado de indicarle lo que debe hacer, reduciendo así un 37,5 % el tiempo de respuesta del flujo de actuación que es lo que suponía consultar al servicio *cloud* (tabla 5.3), y por tanto, proporcionar una mejora de rendimiento sustancial a este caso de uso como se indica en la tabla 5.6 gracias al uso de la computación *edge*. Finalmente, para mantener la consistencia del sistema, el dispositivo *edge* notifica al *cloud* para actualizar el estado que él ha cambiado de manera autónoma. Cabe destacar también dentro de este punto, que el incremento en el uso de la memoria en los dispositivos *edge* es despreciable ya que el tamaño de cada nueva entrada en la caché no excede de 138 bytes, lo que suponiendo un caso de uso intensivo en el que se dispone de 100 señales de tráfico por dispositivo *edge*, el incremento de uso de memoria sería de unos 13,5 KB.

Por otro lado, referido a los dos siguientes puntos, la obtención de la información de la propia vía se realiza sólo una vez y es almacenada en forma de constante en el dispositivo *edge*. Dicha información proporciona al dispositivo la capacidad de modelar el mundo físico mediante teoría de colas, en este caso de uso el mode-

lo es M/M/n/∞. Esto significa que se supone que las llegadas se producen según un proceso de Poisson homogéneo de parámetro λ, que los tiempos de servicio siguen un distribución exponencial de parámetro λ, qu el número de servidores es el número de carriles en este sentido de la carretera (n) y que la capacidad se puede considerar infinita (para dar servicio a todos los dominios observados potenciales, es decir, a todos los coches en la carretera). Por lo tanto, ρ que es el factor de saturación o utilización del sistema se puede calcular como:

$$\rho = \frac{\lambda}{n * \mu}$$

Teniendo en cuenta que el número de servidores es fijo (carriles) en cada segmento, el ratio medio de servicio se puede obtener en base a la velocidad máxima de la vía y a las muestras de telemetría recibidas de los vehículos, y el factor de utilización máximo del sistema viene proporcionado por los servicios *cloud*, el dispositivo *edge* puede realizar el conjunto de observaciones a través de los sensores térmicos para detectar el ratio medio de llegada de peticiones (coches) y poder decidir si requiere algún tipo de acción de control en función de la saturación que obtenga en cada momento comparado con la máxima definida por el sistema sin suponer un coste computacional elevado para el dispositivo.

Finalmente, referido al último punto de la enumeración de funciones adquiridas por los dispositivos *edge*, como cada dispositivo conoce el porcentaje de utilización que tiene tanto su dispositivo predecesor como su posterior, ellos mismos pueden decidir y actuar sobre el mundo físico si el *cloud* permite acciones de control de manera autónoma por parte de los dispositivos *edge* para, entre otras opciones, reducir la velocidad máxima de sus respectivos tramos de carretera o informar de posibles rutas alternativas menos saturadas con el fin de descongestionar los segmentos atascados.

Rol *cloud*

Para dar soporte a la funcionalidad adquirida por los dispositivos *edge* en el caso de *crowdsensing* descrito, el entorno *cloud* de la carretera inteligente debe proporcionar una serie de APIs protegidas por OAuth 2.0 que le facilite a dichos dispositivos la información que requieren en cada momento para llevar a cabo

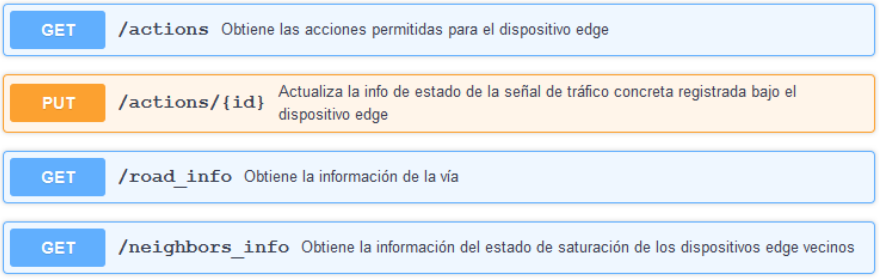

Figura 5.12: APIs *cloud* definidas para el modelo de *crowdsensing*

la computación *edge* planteada en este modelo. En la figura 5.12 se muestra la descripción *swagger* de las APIs creadas a tal efecto.

Con respecto a cuáles son los parámetros de entrada y de salida de cada una de las APIs descritas, a continuación se describe la parametría concreta por cada una de ellas:

- GET */actions*: la función de este servicio consiste en dotar al dispositivo *edge* de la información relacionada con las acciones disponibles para el conjunto de dispositivos IoT que tiene registrados bajo su responsabilidad. Por este motivo, este servicio no tiene ningún tipo de parámetro más allá de estar protegido por OAuth 2.0, es decir, requiere del *token* de acceso del dispositivo *edge*. Gracias a dicho *token* de acceso, el servicio *cloud* identifica qué dispositivo es y realiza la búsqueda de las acciones disponibles para el conjunto de dispositivos IoT que se registraron a través de él. Finalmente, el servicio *cloud* le retorna dicha información al dispositivo *edge* como un enumerado de acciones permitidas para que pueda utilizar dichas acciones disponibles a su voluntad en base a las necesidades que identifique en el tráfico de la vía.

- PUT */actions/{id}*: en el caso de que la calibración a nivel *edge* esté permitida, gracias a este servicio, un dispositivo *edge* que ha modificado el estado de una señal en base a sus cálculos puede notificar al servicio *cloud* con el fin de mantener la consistencia en el sistema. Por este motivo, este servicio recibe como parámetro en el propio *path* de la URL el *token* de identidad que representa al dispositivo IoT que ha cambiado de estado, y como pará-

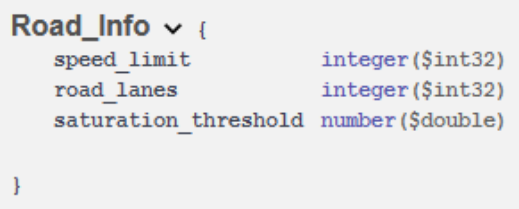

Figura 5.13: Definición del objeto JSON *Road_Info*

metro adicional, la acción que representa el estado actual, el cual ha debido ser elegido de entre los disponibles para el dispositivo *edge*. Al ser un servicio que realiza una modificación en el estado general del sistema, dicha API está protegida por OAuth 2.0 y además, el propio *cloud* verifica que el *token* de identidad proporcionado como identificador esté vinculado con el *token* de acceso del dispositivo *edge* siguiendo de este modo, los requisitos definidos en el esquema de delegación de autorización utilizado. Nótese que la actualización del estado de una señal por parte del dispositivo *edge* sólo se materializa para las señales dentro del dominio de la carretera inteligente, es decir, los dispositivos *edge* de los propios vehículos que actúan a la vez como sensor y como actuador para la carretera inteligente, no generan ningún tipo de persistencia en el *cloud* ya que la telemetría que proporcionan sirve sólo para mejorar la precisión de las recomendaciones y dada su movilidad por la propia carretera, no tiene sentido que la carretera realice una trazabilidad exhaustiva de dichos vehículos.

- GET */road_info*: este servicio le proporciona al dispositivo *edge* la información necesaria y suficiente para poder comenzar a realizar sus cálculos de congestión de tráfico en el segmento de la vía que está desplegado. Cuando el dispositivo consume dicha API, el propio *cloud* identifica qué dispositivo es y en qué segmento está desplegado en base al *token* de acceso de OAuth 2.0 proporcionado. Una vez realizada dicha identificación por parte del *cloud*, éste le devuelve la información de la vía en la que está desplegada el dispositivo *edge* solicitante mediante el objeto *Road_Info* que se muestra en la figura 5.13 y que representa dicha información.

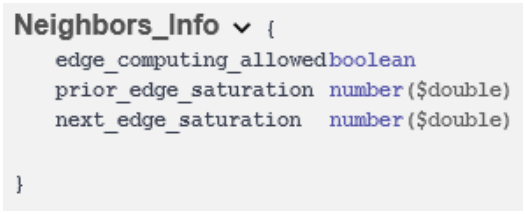

Figura 5.14: Definición del objeto JSON *Neighbors_Info*

- GET */neighbors_info*: finalmente, este último servicio se encarga de enriquecer la computación *edge* llevada a cabo por cada dispositivo al dotarles del dinamismo suficiente para poder tomar decisiones no sólo basadas en sus propios datos recopilados sino añadiendo también, la información de sus dispositivos *edge* vecinos. De la misma forma que para el servicio anterior, el *cloud* identifica el dispositivo, en qué segmento de la vía está desplegado y cuáles son sus vecinos para finalmente, devolver a dicho dispositivo la información referida a los niveles de saturación de sus vecinos y si la computación *edge* está permitida por el *cloud* para llevar a cabo la autocalibración del mundo físico por su cuenta. La figura 5.14 muestra el objeto *Neighbors_Info* devuelto por esta API al dispositivo *edge*.

Cabe destacar que ha quedado fuera del alcance de este diseño los algoritmos y funcionalidades proporcionadas por los servicios *cloud* para aprovechar o enriquecer aún más las decisiones del dominio de aplicación concreto en base a la inmensa cantidad de datos recabados por los diferentes sensores. Esto es así ya que el foco principal del estudio en el caso de uso concreto se ha puesto en la materialización de la solución de *crowdsensing* influenciado apoyado en el esquema de delegación de autorización descrito en capítulos anteriores.

5.3.3. Análisis de privacidad

Con el fin de demostrar las propiedades del esquema planteado desde el punto de vista de preservación de la privacidad, se ha utilizado el modelo de amenazas a la privacidad LINDDUN [184]. Dicho modelo plantea, de manera estructurada, una guía sobre amenazas a la privacidad con el fin de que puedan ser mitigadas

sistémicamente en las diferentes arquitecturas de software.

En la tablas 5.8 y 5.9 se resume el modelo de amenazas formal realizado para abordar los diferentes aspectos sobre el esquema propuesto donde, la primera columna representa la amenaza o riesgo analizado siguiendo el acrónimo LINDDUN, la segunda es el objetivo elegido para materializar la amenaza, la tercera es el ataque concreto y la cuarta, muestra si finalmente el ataque tendría éxito o no sobre el sistema. En el caso de que exista un Sí en la columna de éxito, significa que el atacante es capaz de materializar la amenaza sobre el objetivo concreto usando el ataque correspondiente en la fila. Por otro lado, un No en dicha columna significa que un atacante no tendría éxito al materializar un ataque de esa manera. Finalmente, la quinta columna cualifica la dificultad de materializar dicho ataque en un entorno productivo, siendo los niveles de dificultad: B (bajo), M (medio), y A (alto).

Con el fin de detallar los puntos clave identificados a lo largo del estudio de amenazas a la privacidad realizado, a continuación se lleva a cabo un análisis en profundidad de dichos puntos:

- Vinculación de los eventos generados por los vehículos: teniendo en cuenta que cada vez que un vehículo es contabilizado por su huella térmica dicho evento se genera desde un sensor concreto desplegado en un segmento específico de la carretera, existe la posibilidad de, aún desconociendo la identidad del vehículo, poder vincular a través de la correlación de sus propios eventos el desplazamiento de un vehículo por una carretera. Sin embargo, esta correlación sólo podría llegar a ocurrir dentro del propio contexto efímero de la medición, ya que para el esquema planteado, el contexto de bajo nivel particular de cada medición se desecha en beneficio del estudio de la tendencia o comportamiento de la multitud tanto desde el punto de vista de persistencia como de procesamiento de los datos.

- Identificación de vehículos concretos: la identificación de los propios vehículos dentro del ecosistema de *crowdsensing* influenciado planteado se convierte en un aspecto muy difícil de conseguir. Partiendo de la premisa que la identidad de los vehículos podría estar alojada (o no) en los servicios *cloud* asociados a cada uno de los diferentes vehículos inteligentes y que

Tabla 5.8: Análisis de amenazas basado en el modelo LINDDUN (I)

Amenaza	**Objetivo**	**Ataque**	**Éxito?**	**Dif.**
Vinculación	Entidad	Login vinculante	No	-
		Comunicación no confiable	No	-
		Metadatos en la comunicación	Sí	A
	Canal	Datos en tránsito	No	-
		Contexto de los datos	Sí	A
	Persistencia	Control de acceso débil	No	-
		Protección insuficiente de inferencia	No	-
	Procesamiento	-	No	-
Identificación	Entidad	Login identificable	No	-
		Comunicación no confiable	No	-
		Metadatos en la comunicación	Sí	A
	Canal	Datos en tránsito	No	-
		Contexto de los datos	Sí	A
	Persistencia	Control de acceso débil	No	-
		Anonimización insuficiente	No	-
	Procesamiento	-	No	-
Repudio	Canal	Ofuscación insuficiente	No	-
		Integridad del cifrado débil	No	-
		Uso de MACs insuficiente	No	-
		Manipulación de mensajes	Sí	M
	Persistencia	Integridad del cifrado débil	No	-
		Control de acceso débil a la BD	No	-
		Falta de permisos de edición en la BD	Sí	A
	Procesamiento	-	No	-

Tabla 5.9: Análisis de amenazas basado en el modelo LINDDUN (II)

Amenaza	**Objetivo**	**Ataque**	**Éxito?**	**Dif.**
Detectabilidad	Canal	Canal encubierto débil	Sí	B
		Ataques *side channel*	No	-
		Ocultación débil de info.	No	-
		Insuficiente tráfico *dummy*	Sí	B
		Cobertura de amplio espectro débil	Sí	M
	Persistencia	-	No	-
	Procesamiento	-	No	-
Concienciación	Entidad	Proporcionar demasiada info.	No	-
		Precisión de los datos	No	-

los servicios *cloud* de la propia carretera inteligente no pueden consultar ni compartir información con los primeros, el tipo de sensores desplegados que se plantean en este esquema imposibilita también la compartición de la

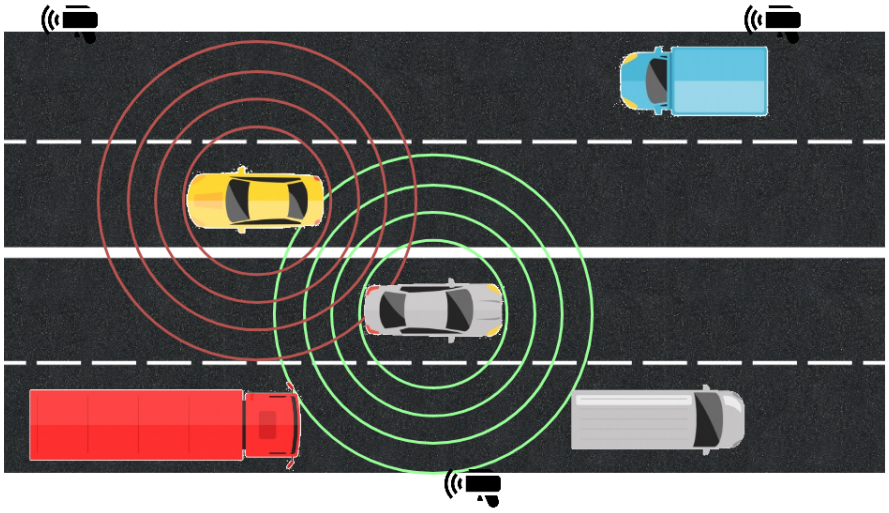

Figura 5.15: Utilización del mundo físico como *firewall*

identidad a través de ellos al convertir el mundo físico en un *firewall* entre cada vehículo (con independencia de si son vehículos autónomos o no disponen de tecnología a bordo) y con la propia carretera como se muestra en la figura 5.15. Por este motivo, la única forma de identificar un vehículo concreto dentro de este esquema se apoya en la calidad de la huella térmica medida y en la información de contexto asociada a la medición para poder determinar la marca o el modelo con el correspondiente coste computacional y la calidad requerida en los sensores utilizados.

- Repudio de los datos: por parte del repudio de los datos recabados, se identifican dos vías principales de ataque al esquema planteado. En primer lugar, la propia manipulación de las mediciones plantea el primer problema, ya sea en el mundo físico debido a errores en la medición por la proximidad extrema entre vehículos que provoca una única lectura individual en lugar de una por vehículo, o por la propia manipulación del mensaje de la medición remitido al dispositivo *edge* correspondiente. Al no estar la medición vinculada con una identidad, la correlación de este tipo de eventualidades maliciosas de manipulación es muy difícil de detectar dentro del sistema pudiendo influir en las decisiones tomadas por el mismo ya que éste parte de

unas muestras incorrectas o manipuladas. Por otro lado, dentro de la base de datos disponible en el servicio *cloud* que soporta el dominio de aplicación en cuestión y de la que se obtienen las recomendaciones, una modificación indebida en las muestras almacenadas derivada de unas insuficientes políticas de control de acceso puede provocar que las decisiones tomadas por el sistema en base a los datos disponibles sean contraproducentes en lugar de beneficiosas para el estado actual del mismo y de los vehículos.

- Detectabilidad: el esquema planteado considera que toda petición en tránsito desde el sensor térmico hasta el dispositivo *edge* para su procesamiento es una petición legítima ya que dado el caso de uso en cuestión, no se puede generar ningún tipo de tráfico *dummy* porque esto conlleva simular la medición de vehículos inexistentes por parte de los sensores. Por este motivo, este esquema adolece de la facilidad para detectar mensajes legítimos en las comunicaciones entre los dispositivos IoT y su dispositivo *edge*. Por otro lado, dada las características físicas que se intentan detectar del entorno, dichas mediciones son susceptibles de la propia interferencia natural del mismo y su grado de impacto en el sistema vendrá definido por la calidad de los sensores desplegados.

- Fuga de información de los servicios *cloud* o dispositivos *edge*: dada la información que recaban los sensores térmicos y que el objetivo del esquema planteado persigue realizar recomendaciones en base a las tendencias de comportamiento y uso de la carretera inteligente por parte de la multitud de vehículos que la utilizan, no se identifica ningún tipo de riesgo de fuga de información sensible o identificable que pueda ser asociada al comportamiento de un vehículo de manera individual.

- Concienciación de los usuarios: al contrario del resto de trabajos de *crowdsensing* en los que es necesario incentivar económicamente a los usuarios para que compartan su información y por ende, es importante concienciarles de las consecuencias de dicha compartición de información, el esquema planteado, al no recabar en ningún momento información personal del usuario, está exento de preocupaciones por parte de los usuarios desde el punto

de vista de si están o no compartiendo demasiada información personal con el sistema. Por otro lado, al no recabar dicha información, ya no es necesario la incentivación económica a los usuarios ya que no se dispone de ningún tipo de información de dichos usuarios que los identifique unívocamente en el sistema.

Finalmente, y en base al análisis de privacidad desglosado a lo largo de los puntos anteriores, se puede concluir que el esquema planteado no gestiona ningún tipo de información sensible sujeta a regulación cuya modificación indebida o extracción malintencionada pudiera derivar en incumplimientos legislativos, regulatorios o de políticas de seguridad de la propia organización que implante dicho esquema.

Capítulo 6

Conclusiones

En este último capítulo se presentan las conclusiones más importante extraídas del trabajo realizado a lo largo de esta tesis doctoral. En primer lugar, se exponen las conclusiones generales. En segundo lugar, se abordan las conclusiones específicas, relativas a los diferentes objetivos planteados para la tesis. Finalmente, se resume el conjunto de líneas de investigación futura que esta tesis doctoral deja abiertas.

6.1. Conclusiones generales

La conclusión principal que se puede extraer a partir del trabajo desarrollado en esta tesis es la validación de la hipótesis de partida expuesta en el primer capítulo. Ha sido posible diseñar mecanismos de seguridad y privacidad para el Internet de las cosas, teniendo en cuenta las características y limitaciones tanto de recursos como de capacidades de comunicación por parte de los dispositivos típicos de estos contextos. Para ello ha sido clave recurrir a arquitecturas de tres capas, en las que dispositivos de borde, ubicados entre los dispositivos IoT y los servicios *cloud*, pueden facilitar soluciones flexibles, escalables y seguras.

Las soluciones para la gestión de identidades y accesos, el direccionamiento y nombrado, o la generación de conocimiento colectivo tradicionales en entornos distribuidos no son aplicables en entornos IoT por los requisitos de los dominios de aplicación habituales y las limitaciones de recursos disponibles (cómputo, al-

macenamiento, comunicaciones y energía) en los dispositivos embebidos en la realidad física.

Por este motivo, ha sido necesario definir nuevas arquitecturas de referencia compatibles con el enfoque *edge-centric* e identificar las tecnologías, estándares y paradigmas que puedan soportar las nuevas soluciones capaces de resolver estos problemas en entornos IoT. En este sentido, la metodología propuesta para desarrollar la presente tesis doctoral ha resultado ser la adecuada para conseguir los objetivos identificados en el capítulo 1 de este documento.

La base de todo el trabajo realizado en esta tesis ha sido el diseño de un esquema de delegación de autorización que ha permitido resolver los problemas tradicionales que implica la gestión de identidades y accesos en IoT. Dicho esquema demuestra que, teniendo en cuenta desde la fase de diseño las características y riesgos que puede suponer la inclusión de dispositivos heterogéneos y con recursos limitados en un sistema altamente escalable y distribuido, los niveles de seguridad del sistema en su conjunto se pueden ver notablemente incrementados. En este sentido, la conclusión fundamental que se puede extraer del trabajo desarrollado es que, teniendo en cuenta la capacidad limitada de cómputo, memoria y almacenamiento de los propios dispositivos, no tiene sentido permitir que accedan a enormes cantidades de datos en los servicios *cloud*, al no disponer de la capacidad suficiente ni para procesarlos ni para almacenarlos, aunque fuera de manera temporal. Además, el establecimiento de un correcto gobierno del dato en el que se le asigna a cada fragmento de información un tiempo de vida en el que es útil para el dispositivo, permite que un usuario malicioso no identifique un retorno de inversión en utilizar el vector de ataque de la suplantación de uno de estos dispositivos con el fin de extraer toda la información disponible en el servicio *cloud*. Dicho de forma sencilla: ajustar correctamente el comportamiento y funcionalidades del sistema en base a los diferentes componentes que lo forman reduce la superficie de exposición, y por ende, los riesgos de seguridad de cada componente.

El esquema propuesto ha permitido plantear un nuevo modelo para el direccionamiento y el nombrado de los dispositivos y garantizar niveles adecuados de privacidad en las aplicaciones que necesitan construir conocimiento colectivo mediante *crowdsensing*.

Desde el punto de vista más práctico o aplicado, los esquemas, modelos y mecanismos propuestos se han implementado en un primer prototipo que se ha probado en dos casos de uso reales (gracias a la colaboración con empresas mediante contratos de investigación e innovación), uno relacionado con la agricultura inteligente y otro con los coches conectados y las carreteras inteligentes. En este aspecto, la principal conclusión es que las soluciones propuestas son realizables y se pueden aplicar en la práctica con independencia de los dispositivos empleados, de los fabricantes o proveedores involucrados, etc. Las tecnologías, estándares y paradigmas que sirven como base a las propuestas realizadas están universalmente extendidos por lo que la integración del prototipo implementado en proyectos reales no ha resultado costosa ni complicada.

Además esta implementación y estos casos de uso han permitido, no sólo validar la funcionalidad de todas las propuestas realizadas a lo largo de la tesis, sino evaluar su rendimiento y los niveles de seguridad y privacidad que permiten alcanzar. Todos ellos adecuados para los dominios de aplicación habituales en la actualidad para IoT.

6.2. Esquemas, modelos y mecanismos propuestos

En el capítulo 3 se ha propuesto el diseño de un esquema de delegación de autorización de dispositivos *edge* a dispositivos con recursos limitados que se apoya en el protocolo OAuth 2.0. Dicho esquema garantiza, sin necesidad de modificar la especificación estándar del protocolo OAuth 2.0 (ya que simplemente añade una serie de características extra sobre él), los niveles de escalabilidad y seguridad necesarios en el ecosistema del Internet de las cosas. Además, el diseño propuesto sienta perfectamente las bases sobre las que proponer un modelo de direccionamiento y nombrado para los dispositivos registrados en el esquema que permite a los servicios *cloud* del dominio de aplicación concreto enviar órdenes a cada dispositivo o grupo de ellos en función de las necesidades existentes en cada momento.

A continuación se resumen las conclusiones más importantes extraídas del trabajo presentado en este capítulo:

- Utilizar los dispositivos *edge* y los enfoques de computación *edge-centric* enriquecen el ecosistema de los dispositivos IoT tanto a nivel de conectividad, como en capacidades de seguridad y tolerancia a fallos.
- La interacción humana es el principal factor limitante a la hora de automatizar procesos en entornos altamente escalables, por lo que dicha interacción debe ser reducida a la mínima expresión o eliminada por completo en esta clase de entornos. Esto es especialmente crítico en el caso de los procedimientos de alta o registro.
- En entornos altamente escalables y distribuidos, es necesario adaptar el nivel de seguridad de los procesos de autenticación al riesgo que implica acceder al conjunto de datos permitido para cada dispositivo.
- La vinculación en los servicios *cloud* de los datos gestionados con la propia identidad que los ha generado y procesa, reduce exponencialmente la superficie de exposición ante la fuga de información en caso de ataques de suplantación de identidad de dispositivos.
- Dotar, no sólo a los *tokens* de acceso a los recursos protegidos, sino al dato en sí mismo de un tiempo de vida útil permite realizar un mejor uso de las capacidades disponibles por parte de los dispositivos con recursos limitados.
- Es necesario resolver aquellos casos en los que los dispositivos tienen movilidad y no siempre están ligados al mismo dispositivo de borde, la solución propuesta debe contemplar algún tipo de *roaming* en la delegación de autorización que permita resolver casos en los que el dispositivo de borde no esté alcanzable desde el dispositivo de recursos limitados.

La definición inicial que se realizó del esquema de delegación de autorización diseñado fue bien recibida en las Jornadas Nacionales SARTECO (en concreto en las Jornadas de Computación Empotrada y Reconfigurable), [185]. Esta buena aceptación y la retroalimentación obtenida impulsó el completo desarrollo del esquema de gestión de identidades y accesos que se presenta en esta tesis doctoral. La definición completa de dicho esquema de delegación de autorización así como su implementación, validación y evaluación de rendimiento y de seguridad

se han publicado en una revista internacional de reconocido prestigio, *Computer Communications* [186].

En relación con la propuesta de modelo de direccionamiento y nombrado, se pueden extraer las siguientes conclusiones:

- El direccionamiento basado en eventos permite garantizar a los diferentes servicios *cloud* la recepción de los mensajes por parte de los dispositivos.
- Aplicar un enfoque *edge-centric* en el modelo de direccionamiento basado en eventos permite independizar a los servicios *cloud* de los protocolos de comunicación ligeros empleados por los dispositivos.
- Al estar tanto los sensores como los actuadores desplegados en el mundo físico para poder obtener información o cambiar su estado respectivamente, el nombrado jerárquico es la mejor solución, al poder aplicar reglas tanto a dispositivos individuales como a grupos de dispositivos o por áreas geográficas.
- Una distribución jerárquica basada en áreas geográficas o de disponibilidad permite a los dominios de aplicación aprovecharse de las capacidades de alta disponibilidad, caché jerárquica y tolerancia a fallos necesarias para dar soporte a los diferentes ecosistemas altamente escalables que plantea el Internet de las cosas.

Estas conclusiones sobre el modelo de direccionamiento y nombrado se han publicado y presentado en el congreso internacional de reconocido prestigio SECRYPT, [187].

Por último hay que destacar las conclusiones relacionadas con el trabajo presentado en el capítulo 4 de esta tesis, es decir, con los mecanismos que permiten garantizar la privacidad de los usuarios cuando se emplean aplicaciones que recurren al conocimiento colectivo:

- El enfoque *edge-centric* es ideal para este tipo de aplicaciones, no sólo por su potencial para establecer mecanismos respetuosos con la privacidad sino por los requisitos en cuanto a latencia o tiempo de respuesta que existen en muchos dominios de aplicación.

- La garantía de niveles adecuados de privacidad mediante validación experimental en casos de uso reales y mediante la generación de un modelo de amenazas completo y exhaustivo puede ayudar mucho a que los usuarios se animen a participar en las aplicaciones de conocimiento colectivo, más allá de otro tipo de alicientes o incentivos que se puedan implementar.
- La colaboración mediante *crowdsensing* influenciado, en el que cada dominio coopera con los demás buscando una solución de equilibrio (en el sentido del equilibrio de la teoría de juegos), ayuda mucho en la consecución de los niveles adecuados de privacidad en estos contextos. La conclusión es que cada dominio recoge información, toma decisiones en función de su estrategia y lo hace maximizando su beneficio conocidas las estrategias del resto de dominios. Esto hace que ninguno de ellos tenga ningún incentivo para modificar individualmente su estrategia (por ejemplo, intentando recoger más información de la necesaria, de dispositivos de otro dominio, etc).

Estas conclusiones se encuentran aún pendientes de publicación en el momento en el que se deposita esta tesis doctoral.

6.3. Prototipo y evaluación

Los esquemas, modelos y mecanismos presentados en los capítulos 3 y 4 se han implementado en un prototipo que se ha probado en dos casos de uso reales. El primero implicaba la monitorización de ganadería vacuna y el segundo, la gestión de carreteras inteligentes. Los detalles de la implementación realizada y de los resultados obtenidos en la validación y evaluación de las propuestas realizadas en esta tesis mediante la realización de diferentes baterías de pruebas y experimentos se han mostrado en el capítulo 5 de este documento. A continuación se resumen las principales conclusiones extraídas:

- Las latencias, consumos de memoria, ancho de banda y energía que implican las soluciones propuestas son adecuadas para los dominios de aplicación habituales de IoT en la actualidad (excluyendo los dominios críticos con requisitos estrictos de tiempo real o necesidad de determinismo).

- Las amenazas para la seguridad que se pueden materializar contra el esquema de delegación de autorización propuesto son pocas, y además sus potenciales impactos no son críticos. Sólo las amenazas a la disponibilidad pueden llegar a suponer un problema en determinados contextos y por este motivo una de las líneas de investigación futura propuestas en la siguiente sección tiene que ver con su mitigación.
- En cuanto a las amenazas a la privacidad que se pueden materializar contra los mecanismos *edge-centric* propuestos para la obtención de conocimiento colectivo, de nuevo han demostrado ser pocas, y en general con impactos poco importantes o muy difíciles de materializar.

Parte de estas conclusiones se ha publicado en los artículos de congreso y de revista mencionados en la sección anterior. Además, la colaboración en proyectos y contratos de investigación e innovación (por ejemplo [188]) para validar las soluciones propuestas en esta tesis han implicado un aprendizaje muy exhaustivo en aspectos tecnológicos como la seguridad de Docker [189] (solución para el despliegue de aplicaciones dentro de contenedores de software que permite virtualizar y aislar las cargas de trabajo que se ejecutan en los dispositivos de borde), de OAuth o de CoAP. Muchas de las conclusiones obtenidas se han mantenido fuera de este documento por considerarse aspectos relacionados con el desarrollo más que con la investigación, pero han permitido la impartición de charlas, talleres y conferencias en diferentes eventos técnicos y de divulgación (por ejemplo [190, 191]). Además, gran parte de los desarrollos realizados (los que no estaban sujetos a diferentes formas de protección de propiedad intelectual o de licencia dada la naturaleza de los casos de uso en los que se ha trabajado) se han compartido como código libre, ya que se espera que se puedan aprovechar para otros proyectos y casos de uso en el futuro fuera del entorno académico y del contexto de esta tesis doctoral [172, 174].

6.4. Líneas de investigación futura

Como ya se ha comentado en las secciones anteriores, la metodología propuesta en el primer capítulo de este documento ha permitido verificar la hipótesis de

partida de la presente tesis doctoral y conseguir los objetivos principales y específicos que se habían propuesto. No obstante, del trabajo realizado surgen nuevas líneas de investigación que en el futuro permitirán completar o mejorar la investigación realizada hasta el momento. En este sentido, algunas de las líneas más interesantes que se han identificado son las siguientes:

- Extender el esquema de delegación de autorización con el fin de soportar otro tipo de protocolos de comunicación ampliamente adoptados en el Internet de las cosas como MQTT (*Message Queue Telemetry Transport*) para las comunicaciones entre los dispositivos IoT y los dispositivos *edge*. Es decir, se trataría de, teniendo en cuenta las características binarias del protocolo MQTT y sin alterar su especificación (como se ha hecho con CoAP en esta tesis), que los dispositivos *edge* sean capaces de intermediar entre las comunicaciones MQTT y las HTTPS con OAuth 2.0, analizando también, el modelo de amenazas de este tipo de traducción.

- Investigar si es posible optimizar el enfoque del direccionamiento basado en eventos planteado utilizando otro tipo de variantes de publicación-suscripción sin perder las características principales del sistema de confirmación de recepción de mensajes por parte de los dispositivos y manteniendo la independencia de los protocolos de comunicación por parte de los servicios *cloud*.

- Proponer un modelo de autenticación adaptativa basada en riesgo que permita a los servicios *cloud* considerar cuándo un dispositivo ha modificado su contexto de forma tan sustancial que realmente debe ser considerado como un nuevo dispositivo, apoyándose para ello, en técnicas de aprendizaje automático e inteligencia artificial. Es decir, proporcionar a los servicios *cloud* los medios suficientes para poder detectar comportamientos anómalos por parte de los dispositivos con el fin de ser capaces de responder de forma inmediata, entre otros, a intentos de ataque de suplantación de identidad.

- Diseñar nuevos mecanismos de tolerancia a fallos con enfoques *edge-centric* con el fin de incrementar la disponibilidad de los dispositivos con recursos limitados y de borde frente a posibles ataques de denegación de servicio

independientemente de su tipo, es decir, ya sean generados tanto de manera deliberada como de manera involuntaria debido a la característica de escalabilidad ya descrita para los contextos IoT.

Hay que destacar que la mayor parte de estas líneas ya se están explorando en nuevos trabajos de investigación o han sido objeto de propuestas de proyectos y contratos de investigación que están pendientes de resolución en el momento en el que se deposita esta tesis doctoral.

Bibliografía

[1] D. Miessler, C. Smith, V. Rudresh, and A. Guzman, "OWASP Internet of Things (IoT) Project," OWASP, Tech. Rep., 2018. [Online]. Available: https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project

[2] H. L'Amrani, B. E. Berroukech, Y. El Bouzekri El Idrissi, and R. Ajhoun, "Identity management systems: Laws of identity for models evaluation," in *4th IEEE International Colloquium on Information Science and Technology (CiSt)*, 2016, pp. 736–740.

[3] D. Hardt, "The OAuth 2.0 Authorization Framework," Internet Requests for Comments, RFC Editor, RFC 6749, October 2012. [Online]. Available: https://www.rfc-editor.org/info/rfc6749

[4] L. Wei and S. Jarzabek, "A Generic Discretionary Access Control System for Reuse Frameworks," in *The Twenty-Second Annual International Computer Software and Applications Conference (Compsac '98) (Cat. No.98CB 36241)*, 1998, p. 356–361.

[5] K. Harsha, B. M. Palavalli, S. Rao, and Ashwin, "Lothlorien: Mandatory Access Control using Linux Security Modules," in *IEEE International Conference on Internet Multimedia Services Architecture and Applications (IMSAA)*, 2009, pp. 1–6.

[6] D. F. Ferraiolo and D. R. Kuhn, "Role-Based Access Controls," in *15th National Computer Security Conference*, 1992, pp. 554–563.

[7] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to Attribute Based Access Control (ABAC) Definition

and Considerations," NIST, Tech. Rep., January 2014. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.sp.800-162.pdf

[8] Organization for the Advancement of Structured Information Standards, "eXtensible Access Control Markup Language (XACML) Version 3.0," OASIS, Tech. Rep., January 2013. [Online]. Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf

[9] X. Jin, R. Sandhu, and R. Krishnan, "RABAC: Role-Centric Attribute-Based Access Control," in *Computer Network Security*, 2012, pp. 84–96.

[10] R. A. C. Diaz, M. Ghita, D. Copot, I. R. Birs, C. Muresan, and C. Ionescu, "Context Aware Control Systems: An Engineering Applications Perspective," *IEEE Access*, vol. 8, pp. 215 550–215 569, 2020.

[11] S. Senthilkumar, M. Viswanatham, and M. Vinothini, "HS-TBAC a highly secured token based access control for outsourced data in cloud," in *International Confernce on Innovation Information in Computing Technologies*, 2015, pp. 1–3.

[12] Y. Zhu, R. Yu, D. Ma, and W. Cheng-Chung Chu, "Cryptographic Attribute-Based Access Control (ABAC) for Secure Decision Making of Dynamic Policy With Multiauthority Attribute Tokens," *IEEE Transactions on Reliability*, vol. 68, no. 4, pp. 1330–1346, 2019.

[13] J. Ahamed and F. Khan, "An Enhanced Context-aware Capability-based Access Control Model for the Internet of Things in Healthcare," in *Sixth HCT Information Technology Trends (ITT)*, 2019, pp. 126–131.

[14] M. Al-Ruithe and E. Benkhelifa, "Cloud Data Governance In-Light of the Saudi Vision 2030 for Digital Transformation," in *IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, 2017, pp. 1436–1442.

[15] G. Thomas, "How to use the DGI data governance framework to configure your program," 2009. [Online]. Avai-

lable: http://www.datagovernance.com/wp-content/uploads/2020/07/wp_how_to_use_the_dgi_data_governance_framework.pdf

[16] M. Al-Ruithe, E. Benkhelifa, Y. Jararweh, and C. Ghedira, "Addressing Data governance in Cloud Storage: Survey, Techniques and Trends," *Journal of Internet Technology*, vol. 19, pp. 1763–1775, 2018.

[17] G. Cheng, Y. Li, Z. Gao, and X. Liu, "Cloud data governance maturity model," in *8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2017, pp. 517–520.

[18] K. A. Saed, N. Aziz, A. W. Ramadhani, and N. Hafizah Hassan, "Data Governance Cloud Security Assessment at Data Center," in *4th International Conference on Computer and Information Sciences (ICCOINS)*, 2018, pp. 1–4.

[19] C. Moon, S. Han, H. Woo, and D. Kim, "Named data networking for infrastructure wireless networks," in *IEEE International Conference on Consumer Electronics (ICCE)*, 2016, pp. 343–344.

[20] Y. N. Rohmah, D. W. Sudiharto, and A. Herutomo, "The performance comparison of forwarding mechanism between IPv4 and Named Data Networking (NDN). Case study: A node compromised by the prefix hijack," in *3rd International Conference on Science in Information Technology (ICSITech)*, 2017, pp. 302–306.

[21] K. Lei, S. Zhong, F. Zhu, K. Xu, and H. Zhang, "An NDN IoT Content Distribution Model With Network Coding Enhanced Forwarding Strategy for 5G," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2725–2735, 2018.

[22] A. Mahmoud, M. Mahyoub, T. Sheltami, and M. Abu-Amara, "Traffic-aware auto-configuration protocol for service oriented low-power and lossy networks in IoT," *Wireless Networks*, pp. 4231–4246, 2019.

[23] M. Pahl, S. Liebald, and C. Lübben, "VSL: A Data-Centric Internet of Things Overlay," in *International Conference on Networked Systems (NetSys)*, 2019, pp. 1–3.

[24] International Telecommunication Union, "ITU Internet Report 2005: The Internet of Things," ITU, Tech. Rep., November 2005. [Online]. Available: http://handle.itu.int/11.1002/pub/800eae6f-en

[25] J. Voas, "Networks of 'Things'," 2016. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-183

[26] International Organization for Standardization, "ISO/IEC 30141:2018 Internet of Things (IoT) - Reference Architecture," 2018. [Online]. Available: https://www.iso.org/standard/65695.html

[27] Alliance for Internet of Things Innovation (AIOTI), "High Level Architecture," 2018. [Online]. Available: https://aioti.eu/wp-content/uploads/2018/06/AIOTI-HLA-R4.0.7.1-Final.pdf

[28] International Telecommunication Union, "Recommendation ITU-T Y.4460: Architectural reference models of devices for Internet of Things applications," 2019. [Online]. Available: https://www.itu.int/rec/dologin_pub.asp?id=T-REC-Y.4460-201906-I!!PDF-E&lang=e&type=items

[29] IEEE, "IEEE Standard for an Architectural Framework for the Internet of Things (IoT)," *IEEE Std 2413-2019*, pp. 1–269, 2020.

[30] E. Borgia, "The Internet of Things vision: Key features, applications and open issues," *Computer Communications*, vol. 54, no. 1, pp. 1–31, 2014.

[31] C. Tsigkanos, S. Nastic, and S. Dustdar, "Towards resilient Internet of Things: Vision, challenges, and research roadmap," in *IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 1754–1764.

[32] J. Guo, I. Chen, and J. J. P. Tsai, "A survey of trust computation models for service management in Internet of Things systems," *Computer Communications*, vol. 97, no. 1, pp. 1–14, 2017.

[33] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 70–95, 2016.

[34] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[35] M. Caprolu, R. Di Pietro, F. Lombardi, and S. Raponi, "Edge Computing Perspectives: Architectures, Technologies, and Open Security Issues," in *IEEE International Conference on Edge Computing (EDGE)*, 2019, pp. 116–123.

[36] E. Marín-Tordera, X. Masip-Bruin, J. García-Almiñana, A. Jukan, G. Ren, and J. Zhu, "Do we all really know what a Fog Node is? Current trends towards an open definition," *Computer Communications*, vol. 109, pp. 117–130, 2017.

[37] R. Fielding and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content," Internet Requests for Comments, RFC Editor, RFC 7231, June 2014. [Online]. Available: https://www.rfc-editor.org/info/rfc7231

[38] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," Internet Requests for Comments, RFC Editor, RFC 7252, June 2014. [Online]. Available: https://www.rfc-editor.org/info/rfc7252

[39] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax," Internet Requests for Comments, RFC Editor, RFC 3986, January 2005. [Online]. Available: https://www.rfc-editor.org/info/rfc3986

[40] N. Freed, J. Klensin, and T. Hansen, "Media Type Specifications and Registration Procedures," Internet Requests for Comments, RFC Editor, RFC 6838, January 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6838

[41] J. Postel, "User Datagram Protocol," Internet Requests for Comments, RFC Editor, RFC 768, August 1980. [Online]. Available: https://www.rfc-editor.org/info/rfc768

[42] Organization for the Advancement of Structured Information Standards, "MQTT Version 5.0," OASIS, Tech. Rep., March 2019. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.pdf

[43] DARPA INTERNET PROGRAM, "Transmission Control Protocol," Internet Requests for Comments, RFC Editor, RFC 793, September 1981. [Online]. Available: https://www.rfc-editor.org/info/rfc793

[44] B. Omoniwa, R. Hussain, M. A. Javed, S. H. Bouk, and S. A. Malik, "Fog/Edge Computing-Based IoT (FECIoT): Architecture, Applications, and Research Issues," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4118–4149, 2019.

[45] A. Rabay'a, E. Schleicher, and K. Graffi, "Fog Computing with P2P: Enhancing Fog Computing Bandwidth for IoT Scenarios," in *International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2019, pp. 82–89.

[46] H. Le, N. Achir, and K. Boussetta, "Fog computing architecture with heterogeneous Internet of Things technologies," in *10th International Conference on Networks of the Future (NoF)*, 2019, pp. 130–133.

[47] A. Kanyilmaz and A. Cetin, "Fog Based Architecture Design for IoT with Private Nodes: A Smart Home Application," in *7th International Istanbul Smart Grids and Cities Congress and Fair (ICSG)*, 2019, pp. 194–198.

[48] M. Yogi, K. Chandrasekhar, and G. Kumar, "Mist Computing: Principles, Trends and Future Direction," *International Journal of Computer Science and Engineering*, vol. 4, pp. 19–21, 2017.

[49] P. Galambos, "Cloud, Fog, and Mist Computing: Advanced Robot Applications," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 6, no. 1, pp. 41–45, 2020.

[50] M. Ejaz, T. Kumar, M. Ylianttila, and E. Harjula, "Performance and Efficiency Optimization of Multi-layer IoT Edge Architecture," in *2nd 6G Wireless Summit (6G SUMMIT)*, 2020, pp. 1–5.

[51] C. Huang, C. Shao, S. Xu, and H. Zhou, "The Social Internet of Thing (S-IOT)-Based Mobile Group Handoff Architecture and Schemes for Proximity Service," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 3, pp. 425–437, 2017.

[52] J. Shafi and A. Waheed, "S-IoT: A new platform for Online Social Networks Using loT," in *1st International Conference on Computer Applications Information Security (ICCAIS)*, 2018, pp. 1–6.

[53] F. Carlier and V. Renault, "IoT-a, Embedded Agents for Smart Internet of Things. Application on a Display Wall," in *IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW)*, 2016, pp. 80–83.

[54] A. R. Biswas and R. Giaffreda, "IoT and cloud convergence: Opportunities and challenges," in *IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 375–376.

[55] E. P. Yadav, E. A. Mittal, and H. Yadav, "IoT: Challenges and Issues in Indian Perspective," in *3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, 2018, pp. 1–5.

[56] S. A. Goswami, B. P. Padhya, and K. D. Patel, "Internet of Things: Applications, Challenges and Research Issues," in *Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2019, pp. 47–50.

[57] N. M. Karie, N. M. Sahri, and P. Haskell-Dowland, "IoT Threat Detection Advances, Challenges and Future Directions," in *Workshop on Emerging Technologies for Security in IoT (ETSecIoT)*, 2020, pp. 22–29.

[58] M. Fazio, R. Ranjan, M. Girolami, J. Taheri, S. Dustdar, and M. Villari, "A Note on the Convergence of IoT, Edge, and Cloud Computing in Smart Cities," *IEEE Cloud Computing*, vol. 5, no. 5, pp. 22–24, 2018.

[59] S. Naveen and M. R. Kounte, "Key Technologies and challenges in IoT Edge Computing," in *Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2019, pp. 61–65.

[60] S. Cirani and M. Picone, "Effective authorization for the Web of Things," in *IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 316–320.

[61] A. Kurniawan and M. Kyas, "A trust model-based Bayesian decision theory in large scale Internet of Things," in *IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2015, pp. 1–5.

[62] S. Sciancalepore, G. Piro, D. Caldarola, G. Boggia, and G. Bianchi, "On the Design of a Decentralized and Multiauthority Access Control Scheme in Federated and Cloud-Assisted Cyber-Physical Systems," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5190–5204, 2018.

[63] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the Internet of Things," *Mathematical and Computer Modelling*, vol. 58, pp. 1189–1205, 2013.

[64] B. Bezawada, K. Haefner, and I. Ray, "Securing Home IoT Environments with Attribute-Based Access Control," in *Proceedings of the Third ACM Workshop on Attribute-Based Access Control*, 2018, pp. 43–53.

[65] H. Kim, E. Kang, D. Broman, and E. Lee, "Resilient Authentication and Authorization for the Internet of Things (IoT) Using Edge Computing," *ACM Transactions on Internet of Things*, vol. 1, no. 1, pp. 1–27, 2020.

[66] A. Alkhresheh, K. Elgazzar, and H. S. Hassanein, "Context-aware Automatic Access Policy Specification for IoT Environments," in *14th Interna-*

tional Wireless Communications Mobile Computing Conference (IWCMC), 2018, pp. 793–799.

[67] Q. Zhou, M. Elbadry, F. Ye, and Y. Yang, "Heracles: Scalable, Fine-Grained Access Control for Internet-of-Things in Enterprise Environments," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 1772–1780.

[68] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, and K. Salah, "A User Authentication Scheme of IoT Devices using Blockchain-Enabled Fog Nodes," in *IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, 2018, pp. 1–8.

[69] M. Wei, E. Liang, and Z. Nie, "A SDN-based IoT Fine-grained Access Control Method," in *International Conference on Information Networking (ICOIN)*, 2020, pp. 637–642.

[70] N. Picard, J. Colin, and D. Zampunieris, "Context-aware and Attribute-based Access Control Applying Proactive Computing to IoT System," in *3rd International Conference on Internet of Things, Big Data and Security (IoTBDS 2018)*, 2018, pp. 333–339.

[71] S. Werner, F. Pallas, and D. Bermbach, *Designing Suitable Access Control for Web-Connected Smart Home Platforms*. Springer International Publishing, 06 2018, pp. 240–251, ISBN: 978-3-319-91763-4.

[72] S. W. Kum, M. Kang, and J. Park, "IoT Delegate: Smart Home Framework for Heterogeneous IoT Service Collaboration," *KSII Transactions on Internet and Information Systems*, vol. 10, pp. 3958–3971, 2016.

[73] A. Ourad, B. Belgacem, and K. Salah, *Using Blockchain for IoT Access Control and Authentication Management*. Springer International Publishing, 06 2018, pp. 150–164, ISBN: 978-3-319-94370-1.

[74] X. Liu, M. Zhao, S. Li, F. Zhang, and W. Trappe, "A Security Framework for the Internet of Things in the Future Internet Architecture," *Future Internet*, vol. 9, no. 3, p. 27, 2017.

[75] S. Sciancalepore, G. Piro, D. Caldarola, G. Boggia, and G. Bianchi, "OAuth-IoT: An access control framework for the Internet of Things based on open standards," in *IEEE Symposium on Computers and Communications (ISCC)*, 2017, pp. 676–681.

[76] J. Batalla, P. Krawiec, M. Gajewski, and K. Sienkiewicz, "ID layer for Internet of Things based on Name-Oriented Networking," *Journal of Telecommunications and Information Technology*, vol. 2013, pp. 40–48, 2013.

[77] H. Moeini, I. Yen, and F. Bastani, "Service Specification and Discovery in IoT Networks," in *IEEE International Conference on Web Services (ICWS)*, 2019, pp. 55–59.

[78] T. Niemirepo, M. Sihvonen, V. Jordan, and J. Heinilä, "Service Platform for Automated IoT Service Provisioning," in *9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2015, pp. 325–329.

[79] J. Sung, "IoT lighting address scheme and profile API design for interoperability," in *International Conference on Information and Communication Technology Convergence (ICTC)*, 2018, pp. 1008–1011.

[80] L. Tarouco, L. Bertholdo, L. Granville, L. Arbiza, F. Carbone, M. Marotta, and J. Santanna, "Internet of Things in healthcare: Interoperatibility and security issues," in *IEEE International Conference on Communications*, 2012, pp. 6121–6125.

[81] G. Tanganelli, C. Vallati, and E. Mingozzi, "Edge-Centric Distributed Discovery and Access in the Internet of Things," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 425–438, 2018.

[82] P. Srisuresh and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," Internet Requests for Comments, RFC Editor, RFC 2663, August 1999. [Online]. Available: https://www.rfc-editor.org/info/rfc2663

[83] L. Lan, F. Li, B. Wang, L. Zhang, and R. Shi, "An Event-Driven Service-Oriented Architecture for the Internet of Things," in *Asia-Pacific Services Computing Conference*, 2014, pp. 68–73.

[84] B. Cheng, D. Zhu, S. Zhao, and J. Chen, "Situation-Aware IoT Service Coordination Using the Event-Driven SOA Paradigm," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 349–361, 2016.

[85] B. Nour, K. Sharif, F. Li, H. Moungla, and Y. Liu, "M2HAV: A Standardized ICN Naming Scheme for Wireless Devices in Internet of Things," in *Wireless Algorithms, Systems, and Applications*, 2017, pp. 289–301.

[86] S. Arshad, B. Shahzaad, M. A. Azam, J. Loo, S. H. Ahmed, and S. Aslam, "Hierarchical and Flat-Based Hybrid Naming Scheme in Content-Centric Networks of Things," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1070–1080, 2018.

[87] M. A. Hail, "IoT-NDN: An IoT Architecture via Named Data Netwoking (NDN)," in *IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, 2019, pp. 74–80.

[88] Z. Yan, N. Kong, Y. Tian, and Y. Park, "A Universal Object Name Resolution Scheme for IoT," in *IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 2013, pp. 1120–1124.

[89] S. Lee, J. Jeong, and J. Park, "DNS Name Autoconfiguration for IoT Home Devices," in *IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, 2015, pp. 131–134.

[90] F. Montori, P. P. Jayaraman, A. Yavari, A. Hassani, and D. Georgakopoulos, "The Curse of Sensing: Survey of techniques and challenges to cope with sparse and dense data in mobile crowd sensing for Internet of Things," *Pervasive and Mobile Computing*, vol. 49, pp. 111 – 125, 2018.

[91] K. Abualsaud, T. M. Elfouly, T. Khattab, E. Yaacoub, L. S. Ismail, M. H. Ahmed, and M. Guizani, "A Survey on Mobile Crowd-Sensing and Its Applications in the IoT Era," *IEEE Access*, vol. 7, pp. 3855–3881, 2019.

[92] Akshat, Gaurav, Zahid, Bhupendra, Aditi, S. Kumar, Maneesha, and P. Pandey, "A Smart Healthcare Monitoring System Using Smartphone Interface," in *4th International Conference on Devices, Circuits and Systems (ICDCS)*, 2018, pp. 228–231.

[93] I. Marin and Z. A. Jabber, "Wireless Sensors Network based on Real-time Healthcare Monitoring," in *International Symposium on Fundamentals of Electrical Engineering (ISFEE)*, 2018, pp. 1–4.

[94] R. G. Utekar and J. S. Umale, "Automated IoT Based Healthcare System for Monitoring of Remotely Located Patients," in *Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1–5.

[95] T. Tamura, "Connected healthcare system to monitor the blood pressure of clients with an unobtrusive device," in *IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2019, pp. 1–6.

[96] S. Sudevan and M. Joseph, "Internet of Things: Incorporation into Healthcare Monitoring," in *4th MEC International Conference on Big Data and Smart City (ICBDSC)*, 2019, pp. 1–4.

[97] . Dilibal, "Development of Edge-IoMT Computing Architecture for Smart Healthcare Monitoring Platform," in *4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2020, pp. 1–4.

[98] J. M. Cecilia, J. Cano, E. Hernández-Orallo, C. T. Calafate, and P. Manzoni, "Mobile crowdsensing approaches to address the COVID-19 pandemic in Spain," *IET Smart Cities*, vol. 2, no. 2, pp. 58–63, 2020.

[99] M. M. Cruz, R. S. Oliveira, A. P. V. Beltrão, P. H. B. Lopes, J. Viterbo, D. G. Trevisan, and F. Bernardini, "Assessing the level of acceptance of

a crowdsourcing solution to monitor infectious diseases propagation," in *IEEE International Smart Cities Conference (ISC2)*, 2020, pp. 1–8.

[100] J. Mohsin, F. H. Saleh, and A. M. Ali Al-muqarm, "Real-time Surveillance System to detect and analyzers the Suspects of COVID-19 patients by using IoT under edge computing techniques (RS-SYS)," in *2nd Al-Noor International Conference for Science and Technology (NICST)*, 2020, pp. 68–73.

[101] M. He, C. Fang, Q. Huang, and J. Yan, "A Remote Monitoring System for Water Quality Based on GPRS in Poor Signal Environment-Poyang Lake for Example," in *26th International Conference on Geoinformatics*, 2018, pp. 1–4.

[102] M. Sheth and P. Rupani, "Smart Gardening Automation using IoT With BLYNK App," in *3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019, pp. 266–270.

[103] A. Hammami, "Smart Environment Data Monitoring," in *International Conference on Computer and Information Sciences (ICCIS)*, 2019, pp. 1–6.

[104] R. Rajalakshmi and J. Vidhya, "Toxic Environment Monitoring Using Sensors Based On Arduino," in *IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2019, pp. 1–6.

[105] Y. Cheng, X. Xu, Y. Du, P. Guan, S. Liu, and L. Zhao, "Design of Air Quality Monitoring System Based on NB-IoT," in *IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, 2019, pp. 385–388.

[106] D. Pujara, P. Kukreja, and S. Gajjar, "Design and Development of E-Sense: IoT based Environment Monitoring System," in *IEEE Students Conference on Engineering Systems (SCES)*, 2020, pp. 1–5.

[107] A. Kazmi, E. Tragos, and M. Serrano, "Underpinning IoT for Road Traffic Noise Management in Smart Cities," in *IEEE International Conference*

on Pervasive Computing and Communications Workshops (PerCom Workshops), 2018, pp. 765–769.

[108] J. AbeBer, M. Gotze, S. Kuhnlenz, R. Grafe, C. Kuhn, T. ClauB, and H. Lukashevich, "A Distributed Sensor Network for Monitoring Noise Level and Noise Sources in Urban Environments," in *IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2018, pp. 318–324.

[109] A. Ghosh, K. Kumari, S. Kumar, M. Saha, S. Nandi, and S. Saha, "NoiseProbe: Assessing the Dynamics of Urban Noise Pollution through Participatory Sensing," in *11th International Conference on Communication Systems Networks (COMSNETS)*, 2019, pp. 451–453.

[110] Y. Liu, X. Ma, L. Shu, Q. Yang, Y. Zhang, Z. Huo, and Z. Zhou, "Internet of Things for Noise Mapping in Smart Cities: State of the Art and Future Directions," *IEEE Network*, vol. 34, no. 4, pp. 112–118, 2020.

[111] T. Ramburn, D. Badoreea, and S. Cheerkoot-Jalim, "DriveMU: A Real-time Road-Traffic Monitoring Android Application for Mauritius," in *Conference on Next Generation Computing Applications (NextComp)*, 2019, pp. 1–8.

[112] M. Raza, A. R. Barket, A. U. Rehman, A. Rehman, and I. Ullah, "Mobile Crowdsensing based Architecture for Intelligent Traffic Prediction and Quickest Path Selection," in *International Conference on UK-China Emerging Technologies (UCET)*, 2020, pp. 1–4.

[113] T. Ludwig, T. Siebigteroth, and V. Pipek, "CrowdMonitor: Monitoring Physical and Digital Activities of Citizens During Emergencies," in *International Conference on Social Informatics*, 2015, pp. 421–428.

[114] S. Minson, B. Brooks, C. Glennie, J. Murray, J. Langbein, S. Owen, T. Heaton, B. Iannucci, and D. Hauser, "Crowdsourced Earthquake Early Warning," *Science Advances*, vol. 1, 2015.

[115] X. Wang, J. Zhang, X. Tian, X. Gan, Y. Guan, and X. Wang, "Crowdsensing-Based Consensus Incident Report for Road Traffic Acqui-

sition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2536–2547, 2018.

[116] C. Wang, Z. Xie, L. Shao, Z. Zhang, and M. Zhou, "Estimating Travel Speed of a Road Section Through Sparse Crowdsensing Data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3486–3495, 2019.

[117] H. Wu, Y. Yu, S. Qian, and D. Tao, "Crowdsensing based Real-time Traffic Condition Assessment Method," in *IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*, 2020, pp. 1–2.

[118] A. S. El-Wakeel, J. Li, A. Noureldin, H. S. Hassanein, and N. Zorba, "Towards a Practical Crowdsensing System for Road Surface Conditions Monitoring," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4672–4685, 2018.

[119] Y. Jeng, S. Huang, and C. Lai, "Inspect Road Quality by Using Anomaly Detection Approach," in *International Conference on System Science and Engineering (ICSSE)*, 2018, pp. 1–4.

[120] Y. Yuan and X. Che, "Research on Road Condition Detection Based on Crowdsensing," in *IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 2019, pp. 804–811.

[121] Y. Wei and F. Gao, "Architecture Design Method for Structural Health Monitoring System (SHM) of Civil Aircraft," in *International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, 2017, pp. 736–739.

[122] Y. Shi, Y. Zhao, R. Xie, and G. Han, "Designing a Structural Health Monitoring System for the Large-scale Crane with Narrow Band IoT," in *IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2019, pp. 239–242.

[123] H. Yao, K. Lin, I. Ungurean, and Y. Yang, "Constrained Relay Node Deployment in Wireless Sensor Network for Structural Health Monitoring," in *International Conference on Sensing and Instrumentation in IoT Era (ISSI)*, 2019, pp. 1–5.

[124] Z. Wang, X. Pang, J. Hu, W. Liu, Q. Wang, Y. Li, and H. Chen, "When Mobile Crowdsensing Meets Privacy," *IEEE Communications Magazine*, vol. 57, no. 9, pp. 72–78, 2019.

[125] I. J. Vergara-Laurens, L. G. Jaimes, and M. A. Labrador, "Privacy-Preserving Mechanisms for Crowdsensing: Survey and Research Challenges," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 855–869, 2017.

[126] Y. Liu, L. Kong, and G. Chen, "Data-Oriented Mobile Crowdsensing: A Comprehensive Survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2849–2885, 2019.

[127] P. Zhou, W. Chen, S. Ji, H. Jiang, L. Yu, and D. Wu, "Privacy-Preserving Online Task Allocation in Edge-Computing-Enabled Massive Crowdsensing," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7773–7787, 2019.

[128] J. Xiong, R. Ma, L. Chen, Y. Tian, Q. Li, X. Liu, and Z. Yao, "A Personalized Privacy Protection Framework for Mobile Crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.

[129] G. Sun, S. Sun, H. Yu, and M. Guizani, "Toward Incentivizing Fog-Based Privacy-Preserving Mobile Crowdsensing in the Internet of Vehicles," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4128–4142, 2020.

[130] Q. Xu, Z. Su, M. Dai, and S. Yu, "APIS: Privacy-Preserving Incentive for Sensing Task Allocation in Cloud and Edge-Cooperation Mobile Internet of Things With SDN," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5892–5905, 2020.

[131] X. Ma, W. Deng, F. Wang, M. Hu, F. Chen, and M. M. Hassan, "TIMCC: On Data Freshness in Privacy-Preserving Incentive Mechanism Design for

Continuous Crowdsensing Using Reverse Auction," *IEEE Access*, vol. 8, pp. 1777–1789, 2020.

[132] X. Wang, S. Garg, H. Lin, G. Kaddoum, J. Hu, and M. S. Hossain, "PPCS: An Intelligent Privacy-Preserving Mobile Edge Crowdsensing Strategy for Industrial IoT," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[133] L. Li, D. Shi, X. Zhang, R. Hou, H. Yue, H. Li, and M. Pan, "Privacy Preserving Participant Recruitment for Coverage Maximization in Location Aware Mobile Crowdsensing," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.

[134] C. Zhang, L. Zhu, C. Xu, J. Ni, C. Huang, and X. S. Shen, "Efficient and Privacy-Preserving Non-Interactive Truth Discovery for Mobile Crowdsensing," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

[135] S. Zhang, S. Ray, R. Lu, Y. Zheng, and J. Shao, "Preserving Location Privacy for Outsourced Most-Frequent Item Query in Mobile Crowdsensing," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[136] X. Zhang, L. Liang, C. Luo, and L. Cheng, "Privacy-Preserving Incentive Mechanisms for Mobile Crowdsensing," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 47–57, 2018.

[137] H. Tschofenig and T. Fossati, "Transport Layer Security (TLS) / Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things," Internet Requests for Comments, RFC Editor, RFC 7925, July 2016. [Online]. Available: https://www.rfc-editor.org/info/rfc7925

[138] Organization for the Advancement of Structured Information Standards, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0," OASIS, Tech. Rep., March 2005. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[139] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore, "OpenID Connect Core 1.0 incorporating errata set 1," OpenID Foundation, Tech. Rep., November 2014. [Online]. Available: https://openid.net/specs/openid-connect-core-1_0.html

[140] Y. Targali, V. Choyi, and Y. Shah, "Seamless authentication and mobility across heterogeneous networks using federated identity systems," in *IEEE International Conference on Communications Workshops (ICC)*, 2013, pp. 1232–1237.

[141] European Parliament, "Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015 on payment services in the internal market, amending Directives 2002/65/EC, 2009/110/EC and 2013/36/EU and Regulation (EU) No 1093/2010, and repealing Directive 2007/64/EC (Text with EEA relevance)," November 2015. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32015L2366

[142] The Berlin Group, "PSD2 Access to Bank Accounts)," 2017. [Online]. Available: https://www.berlin-group.org/psd2-access-to-bank-accounts

[143] M. Atwood, R. M. Conlan, B. Cook, L. Culver, K. Elliott-McCrea, L. Halff, E. Hammer-Lahav, B. Laurie, C. Messina, J. Panzer, S. Quigley, D. Recordon, E. Sandler, J. Sergent, T. Sieling, B. Slesinsky, and A. Smith, "OAuth Core 1.0," OAuth Core Workgroup, Tech. Rep., December 2007. [Online]. Available: https://oauth.net/core/1.0/

[144] M. Atwood, D. Balfanz, D. Bounds, R. M. Conlan, B. Cook, L. Culver, B. de Medeiros, B. Eaton, K. Elliott-McCrea, L. Halff, E. Hammer-Lahav, B. Laurie, C. Messina, J. Panzer, S. Quigley, D. Recordon, E. Sandler, J. Sergent, T. Sieling, B. Slesinsky, and A. Smith, "OAuth Core 1.0 Revision A," OAuth Core Workgroup, Tech. Rep., June 2009. [Online]. Available: https://oauth.net/core/1.0a/

[145] OAuth Community, "OAuth Security Advisory 2009.1: A session fixation attack against the OAuth Request Token approval flow (OAuth Core

1.0 Section 6) has been discovered," April 2009. [Online]. Available: https://oauth.net/advisories/2009-1/

[146] T. Lodderstedt, M. McGloin, and P. Hunt, "OAuth 2.0 Threat Model and Security Considerations," Internet Requests for Comments, RFC Editor, RFC 6819, January 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6819

[147] J. Richer, "OAuth 2.0 Token Introspection," Internet Requests for Comments, RFC Editor, RFC 7662, October 2015. [Online]. Available: https://www.rfc-editor.org/info/rfc7662

[148] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," Internet Requests for Comments, RFC Editor, RFC 7519, May 2015. [Online]. Available: https://www.rfc-editor.org/info/rfc7519

[149] S. Cirani, M. Picone, P. Gonizzi, L. Veltri, and G. Ferrari, "IoT-OAS: An OAuth-Based Authorization Service Architecture for Secure Services in IoT Scenarios," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 1224–1234, 2015.

[150] P. Solapurkar, "Building secure healthcare services using OAuth 2.0 and JSON web token in IoT cloud scenario," in *2nd International Conference on Contemporary Computing and Informatics (IC3I)*, 2016, pp. 99–104.

[151] F. Fernández, A. Alonso, L. Marco, and J. Salvachúa, "A model to enable application-scoped access control as a service for IoT using OAuth 2.0," in *20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, 2017, pp. 322–324.

[152] F. Chen, Y. Luo, J. Zhang, J. Zhu, Z. Zhang, C. Zhao, and T. Wang, "An Infrastructure Framework for Privacy Protection of Community Medical Internet of Things," *World Wide Web*, vol. 21, no. 1, p. 33–57, 2018.

[153] M. Beltrán, "Identifying, authenticating and authorizing smart objects and end users to cloud services in Internet of Things," *Computers & Security*, vol. 77, pp. 595–611, 2018.

[154] L. Arnaboldi and H. Tschofenig, "A Formal Model for Delegated Authorization of IoT Devices Using ACE-OAuth," in *4th OAuth Security Workshop 2019 (OSW 2019)*, 2019.

[155] D. Lagutin, Y. Kortesniemi, N. Fotiou, and V. A. Siris, "Enabling Decentralised Identifiers and Verifiable Credentials for Constrained Internet-of-Things Devices using OAuth-based Delegation," in *Proceedings of the Workshop on Decentralized IoT Systems and Security (DISS)*, 2019.

[156] F. Tehranipoor, N. Karimian, W. Yan, and J. A. Chandy, "DRAM-Based Intrinsic Physically Unclonable Functions for System-Level Security and Authentication," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 3, pp. 1085–1097, 09 2017.

[157] M. N. Aman, K. C. Chua, and B. Sikdar, "Mutual Authentication in IoT Systems Using Physical Unclonable Functions," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1327–1340, 2017.

[158] S. Akkermans, W. Daniels, G. Sankar R., B. Crispo, and D. Hughes, "CerberOS: A Resource-Secure OS for Sharing IoT Devices," in *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*, 2017, pp. 96–107.

[159] H. Kim and E. A. Lee, "Authentication and Authorization for the Internet of Things," *IT Professional*, vol. 19, no. 5, pp. 27–33, 2017.

[160] G. Aliberti, R. Pietro, and S. Guarino, "SLAP: Secure Lightweight Authentication Protocol for Resource-constrained Devices," in *Proceedings of the 5th International Conference on Security and Cryptography*, 2017, pp. 163–174.

[161] W. Denniss, J. Bradley, M. Jones, and H. Tschofenig, "OAuth 2.0 Device Authorization Grant," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-oauth-device-flow-15, March 2019. [Online]. Available: https://tools.ietf.org/html/draft-ietf-oauth-device-flow-15

[162] E. Maler, M. Machulak, and D. Catalano, "User-Managed Access (UMA) Profile of OAuth 2.0," Kantara Initiative, Tech. Rep., December 2015. [Online]. Available: https://docs.kantarainitiative.org/uma/draft-uma-core.html

[163] T. Levä, O. Mazhelis, and H. Suomi, "Comparing the cost-efficiency of CoAP and HTTP in Web of Things applications," *Decision Support Systems*, vol. 63, pp. 23 – 38, 2014.

[164] S. Aneja, N. Aneja, and M. S. Islam, "IoT Device Fingerprint using Deep Learning," in *IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, 2018, pp. 174–179.

[165] K. Yang, Q. Li, and L. Sun, "Towards automatic fingerprinting of IoT devices in the cyberspace," *Computer Networks*, vol. 148, pp. 318–327, 2019.

[166] International Telecommunication Union, "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks," ITU, Tech. Rep., March 2000. [Online]. Available: https://www.itu.int/rec/T-REC-X.509-201910-I/en

[167] C. Adams, S. Farrell, T. Kause, and T. Mononen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)," Internet Requests for Comments, RFC Editor, RFC 4210, September 2005. [Online]. Available: https://www.rfc-editor.org/info/rfc4210

[168] P. Mockapetris, "Domain Names - Concepts and Facilities," Internet Requests for Comments, RFC Editor, RFC 1034, November 1987. [Online]. Available: https://www.rfc-editor.org/info/rfc1034

[169] G. Wachob, D. Reed, M. LeMaitre, D. McAlpin, and D. McPherson, "XRI requirements and glossary," OASIS, Tech. Rep., June 2003. [Online]. Available: http://xml.coverpages.org/XRI-REQv110.pdf

[170] D. van Thuan, P. Butkus, and D. van Thanh, "A User Centric Identity Management for Internet of Things," in *International Conference on IT Convergence and Security (ICITCS)*, 2014, pp. 1–4.

[171] EventHelix.com Inc., "Queueing Theory Basics." [Online]. Available: http://www.eventhelix.com/realtimemantra/congestioncontrol/queueing_theory.htm

[172] E. Grande and M. Beltrán (a), "Edge-Centric Delegation of Authorization for Constrained Devices in Internet of Things - Cloud Role Source Code," 2019. [Online]. Available: https://bitbucket.org/egrander/urjc_poc_001_cloud

[173] G. Tanganelli, C. Vallati, and E. Mingozzi, "CoAPthon: Easy development of CoAP-based IoT applications with Python," in *IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 63–68.

[174] E. Grande and M. Beltrán (b), "Edge-Centric Delegation of Authorization for Constrained Devices in Internet of Things - IoT Role Source Code," 2019. [Online]. Available: https://bitbucket.org/egrander/urjc_poc_001_iot

[175] Microsoft, "The STRIDE Threat Model," 2009. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)?redirectedfrom=MSDN

[176] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, "STRIDE-based threat modeling for cyber-physical systems," in *IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, 2017, pp. 1–6.

[177] Y. Iwasaki, M. Misumi, and T. Nakamiya, "Robust Vehicle Detection under Various Environments to Realize Road Traffic Flow Surveillance Using an Infrared Thermal Camera," *The Scientific World Journal*, vol. 2015, 2015.

[178] C. Chang, K. Srinivasan, Y. Chen, W. Cheng, and K. Hua, "Vehicle Detection in Thermal Images Using Deep Neural Network," in *IEEE Visual Communications and Image Processing (VCIP)*, 2018, pp. 1–4.

[179] N. Sangeetha, K. Sathyanarayan, and A. A. Kadar, "Vehicle Detection using Thermal Sensors," *International Journal of Engineering Research & Technology (IJERT)*, vol. 08, 2019. [Online]. Available: https://www.ijert.org/vehicle-detection-using-thermal-sensors

[180] V. Q. Dinh, Y. Lee, H. Choi, and M. Jeon, "Real-Time Traffic Sign Recognition," in *IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, 2018, pp. 206–212.

[181] M. K. Hasan, M. Shahjalal, M. Z. Chowdhury, N. Tuan Le, and Y. M. Jang, "Simultaneous Traffic Sign Recognition and Real-Time Communication using Dual Camera in ITS," in *International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2019, pp. 517–520.

[182] Y. Sun, P. Ge, and D. Liu, "Traffic Sign Detection and Recognition Based on Convolutional Neural Network," in *Chinese Automation Congress (CAC)*, 2019, pp. 2851–2854.

[183] M. T. Islam, "Traffic sign detection and recognition based on convolutional neural networks," in *International Conference on Advances in Computing, Communication and Control (ICAC3)*, 2019, pp. 1–6.

[184] K. Wuyts, R. Scandariato, and W. Joosen, "LINDDUN Privacy Threat Modeling," 2015. [Online]. Available: https://www.linddun.org/

[185] E. Grande and M. Beltrán, "Delegación de Autorización Perimetral para Dispositivos IoT," in *Avances en Arquitectura y Tecnología de Computadores. Actas de las Jornadas SARTECO 2019. Cáceres, 18 a 20 de septiembre de 2019*, 2019, pp. 641–648.

[186] E. Grande and M. Beltrán, "Edge-centric delegation of authorization for constrained devices in the Internet of Things," *Computer Communications*, vol. 160, pp. 464 – 474, 2020.

[187] E. Grande and M. Beltrán, "Securing Device-to-Cloud Interactions in the Internet of Things Relying on Edge Devices," in *17th International Joint Conference on e-Business and Telecommunications - Volume 3: SECRYPT*, 2020, pp. 559–564.

[188] "An infrastructure for highly decentralized hybrid systems EDGEDATA," 2021. [Online]. Available: https://www:networks:imdea:org/research/projects/edgedata-cm

[189] F. Ramírez, E. Grande, and R. Troncoso, *Docker: SecDevOps*. 0xWORD, 2018, ISBN: 978-84-697-9752-5.

[190] E. Grande (a), "Docker & SecDevOps," 2018. [Online]. Available: https://www.navajanegra.com/2018/agenda-talleres/

[191] E. Grande (b), "OAuth 2 en la Era del IoT," 2020. [Online]. Available: https://www.youtube.com/watch?v=reIK6XERnKs