Escuela Politécnica Superior, Universidad Carlos III de Madrid

Trabajo de Fin de Grado

Grado en Ingeniería en Tecnologías de Telecomunicación

ESTUDIO DE UN PROTOCOLO DE IDENTIFICACIÓN PARA EL INTERNET DE LAS COSAS

Autor: Kevin Padilla Cañadas

Director: Dr. Daniel Díaz-Sánchez

15 de octubre de 2015

Página en blanco.

Resumen

Internet de las Cosas, que nos traerá la unión entre el mundo físico y el mundo digital, se encuentra en fase de desarrollo, pero las primeras muestras de *cosas* conectadas que hemos podido ver dejan en entredicho la seguridad y la privacidad de los usuarios. Este proyecto aborda el tema y proporciona a IoT mecanismos de seguridad en el control de acceso basados en identidad digital usando tecnología OpenID y desarrolla una plataforma de simulación basada en el modelo de actores e implementada con Akka y cuyo propósito es simular el funcionamiento de OpenID en entornos de *cosas* reales.

Abstract

Internet of Things, which will bring the union between physical world and digital world, is now in development phase, but first samples of connected *things* that we have seen cast serious doubts on security and users'privacy. This project address that issue and provides IoT with security mechanisms for access control, based on digital identity using OpenID technology, and develops a simulation platform, based on actor model and implemented with Akka, that has as purpose to simulate OpenID functioning over real environments of *things*.

"Yo no tengo sueños, tengo objetivos.
Pasemos a lo siguiente."
Harvey Specter

Índice

Índice de figuras

Página en blanco.

1. Introduction

In last years, it has sound stronger the concept of *Smart Environments*: Smart Cities, Smart Grid, Smart Home, Smart Mobility, Smart Building... Though at this moment these concepts are in development, it is a matter of time that we start seeing these Smart Environments to spread.

The way we currently face day-to-day and the way we interact with daily objects will substantially change with the integration of electronic devices with the real world. Environment sensorization and device hyperconnectivity shape the so-called *Internet of Things*, *IoT* or *Internet of Everything*, in which things are aware of its self situation and the state of its surroundings, and they take intelligent decisions to get a particular goal [1].

Internet of Things ecosystem is rapidly emerging. According to expert estimations, it is expected that more than 50 billion devices will be connected by 2020, following an exponential growth [2]. The number of connections between devices and especially machine-to-machine communication (M2M) will grow proportionally. That represent an actual challenge for the current network communications and the topology of future networks has to be called into question. Distributed systems, highly decentralized and extremely flexible to adapt coming changes will be essential requirements for the deployment of the Internet of Things.

The huge amount of data generated by sensors and devices will be target of *Big Data* and *Cloud Computing* to extract useful and usable information to improve quality and efficiency of services by means of data analysis. IoT will be the main core of economy and industries so as to reduce costs, optimize the use of resources, improve efficiency and increase productivity by adding value across all other industries.

However, though nobody doubt about the incredible potential of IoT in a future, it still have to mature and overtake many challenges to settle its growth. Current deployment of IoT environments is characterized by following a vertical structure and using proprietary software that limit interoperability with the rest of ecosystems [3]. The lack of consensus about protocols and standards and the uncertainty about the direction that IoT evolution will take do not make spreading easy.

On the other hand, issues with respect to privacy and security have to be considered. In an environment full sensors and devices gathering and processing information of any type constantly, it is exceptionally important to guarantee that data is kept beyond non-authorized third party reach, either people or things, and guarantee user's privacy and an ethic approved usage of the information.

This project focus on the problem of security in the Internet of Things, specifically on things identification issues. Devices ought to be able to distinguish between allowed and non-allowed communications, and be able to identify others and identify oneself to others in order to interact with the environment in a reliable

way and guarantee a secure data exchange.

1.1. Description of the problem

Internet of Things has already arrived, although it is in a premature phase, and one of the major weaknesses of devices that start to shape IoT is security. A research carried out by HP in July 2014 [4] analyzed 10 of the most common IoT devices like TVs, thermostats, smart plugs, irrigation systems, intelligent locks or multi-device control hubs, and the results showed an alarming number of vulnerabilities.

80 % of these devices gathered personal information, and many of them transmitted such information through local network and Internet without any kind of encryption, which suppose a high risk for privacy when transmitted information is sensitive and confidential. Moreover, also the 80 % of analyzed devices presented any kind of vulnerability due to weak password requirements and insufficient authentication and authorization mechanisms.

These debilites are clearly manifested in this article of *El País* [5], in which Sophos company demonstrate how easy is to get access control of connected devices. An attacker that get access to a WiFi network can get complete control over things connected to the same local network. Just by executing a script, the attacker can turn on and off a light bulb without any kind of authentication or identity control. This kind of vulnerabilities can present a high risk if devices connected to the network are more relevant or when they control key environment systems. For example, it could be a risk that the access to the heating or to the security alarm of a house was as easy as shown in the article. In other environments like an Smart City, it would be an actual danger if the control system that regulates traffic lights in a crossroads were compromised in such way; or if Smart Grid devices were altered as easy as light bulb, it could suppose loses for an electric company.

1.2. Motivation

This project comes up on account of the obvious deficiency on security services for the Internet of Things in all its points: confidentiality, integrity, authenticity and access control.

As it has been proved, there is much pending work in the field of security for Internet of Things. Fortunately, there is still some time leeway, since it is not expected until 2020 to reach 50 billion connected devices that will gather and share information and will control multiple aspects of our life and environment. It is important to establish a solid and secure base to settle the technology and communication and control protocols that we will see and use for the next decades.

The motivation of this project is to meet the need of security that Internet of

Things has right now, since its a great barrier for its deployment and expansion, besides the fact that it is a risk for users privacy and society security. It is crucial that future things and connected devices are able to guarantee digital security.

1.3. Objectives

In a society where Information Technologies have more and more importance, it is possible to establish two objectives in broad strokes. On one hand, a social objective for raising awareness and educating on Information Technologies. On the other hand, a technical objective to improve security services and communication protocols for Internet of Things. Both objectives are necessary and are complementary with each other.

Social education in Information Technologies started with the newest generations a decade ago, with the arrival of new technologies. However, the awareness for security is not shown in the adoption of new technologies to everyday life. There exist much lack of information and education on technologies, what drives average users to be in constant risk in the Internet. Usual actions like surfing the Web and choosing a password can take a user to put its privacy in danger. Thus, it is necessary that authorities take part to reinforce security education on Information Technology, a basic pillar of modern society.

On the other hand, providing devices with security services is very important to build *Smart* hyperconnected reliable environments. The whole network must accomplish with the security requirements to store, transmit and process information [6] [7]:

- Confidentiality: prevent information to be accessible by non-authorized third parties.
- Authenticity: guarantee that the origin of information is appropriately identified.
- Integrity: guarantee that non-authorized third parties have not modified transmitted information.
- Availability: assure that information is available when authorized entities request it.
- Access control: regulate the access to systems and information.
- Non-repudiation: guarantee that neither emitter nor receptor can deny the transmission of information.

Besides, due to the great scale that Internet of Things will have, any system or security protocol must accomplish with some additional requirements:

- Scalable: ensure the proper functioning given a growth of the network.

- Robustness: guarantee the proper functioning given a network failure, a transmission failure or wrong data.
- Flexibility: ability to adapt to new devices with different functionality, capacity and characteristics.
- Lightness: any device should be able to implement basic security functionalities regardless of its computational power.

As explained before, Internet of Thing will integrate electronic devices within physical world, allowing people to digitally interact with our surroundings, and things to interact between them. However, it is possible that access to some devices shall be restricted to a group of users or modeled to allow different type of accesses for different roles. For example, a home thermostat can be modified by its owners Bob and Alice, but their son Mike only can modify the temperature of his bedroom just 2°C up or down; guests only have access to thermostat to check the temperature of the house, but they can not modify anything without owners explicit permission; moreover, electric company has access to thermostat to gather data about consume habits of the house and, in case of failure, a technician can access to the settings. Thus, an IoT environment must be multi-user and multi-role to support this kind of scenario.

1.3.1. Objectives of the project

Due to time and resources limitations for this project, it would be quite difficult to accomplish every objective explained previously. Therefore, this project will focus on *access control*.

The aim of this project is to provide the Internet of Things environments with mechanisms that allow for establishing a policy on access control to information and devices. For that, *digital identity* technology will be used to resolve security deficiencies of access control so that, when an attacker command a light bulb to switch off, it ask for identification and authorization before anything (section B.1, [5]).

Considering that implementing a digital identity system in a real IoT environment is beyond the possibilities of this project because of, again, time and resources, it will be developed a software simulation platform. The aim of this simulator is to be able to test digital identity functioning on simulated environments. Furthermore, new functionalities or more exhaustive tests could be checked in a future on this simulator with different testing environments to measure load, performance, cost or any parameter of interest.

2. Estado del arte

2.1. Concepto de Internet de las Cosas

El *Internet de las Cosas* o *Internet of Things*, también llamado *Internet of Everything* [8] es un concepto que se refiere a la integración del mundo físico con Internet mediante la conexión digital de objetos o *cosas*. El término de *Internet de las Cosas* fue acuñado en 1999 por Kevin Ashton, cofundador y director ejecutivo de *Auto-ID Center* del Massachusetts Institute of Technology (MIT), donde se desarrolló la tecnología de identificación por radiofrecuencia (RFID) [9].

IoT representa la convergencia y unificación de varias tecnologías ya desarrolladas [8]:

- RFID (*Radio-Frequency Identification*): tecnología de identificación automática de objetos que, a través de ondas electromagnéticas, leen la información almacenada en etiquetas RFID.
- Comunicación M2M (*machine-to-machine*): tecnología que permite la comunicación entre dispositivos de forma autónoma.
- Computación ubicua: concepto en el que la computación rodea el entorno de la persona, pudiendo interactuar con objetos cotidianos.
- WoT (*Web of Things*): término que describe la utilización de los estándares actuales de la Web para integrar y acceder en la red a objetos reales con dispositivos electrónicos embebidos.

Estas tecnologías han sido utilizadas en la última década de forma separada para el desarrollo de soluciones privativas para crear sistemas conectados o inteligentes en los sectores de la automoción y el transporte (telepeaje, parquímetros, información de tráfico...), domótica (electrónica de consumo, automatización, contadores inteligentes, seguridad) y sanidad (monitorización, diagnóstico y prevención). Esto ha dado lugar a silos verticales independientes e incompatibles entre ellos, con un nivel de estandarización muy bajo, lo que conlleva una gran limitación para conseguir una mayor penetración en los mercados.

En IoT, una *cosa* es un objeto físico o virtual, inteligente y con capacidad de comunicación con los demás objetos y dispositivos, por una o varias interfaces de comunicación, inalámbricas o no, y capacidad para procesar mensajes e información. Cada *cosa* puede tener una o varias finalidades definidas, como *sensorizar* variables del entorno físico en el que se encuentra localizada, reaccionar ante información sentida o recibida de otras cosas o enviar la información a otros dispositivos o cosas.

2.2. Seguridad en el Internet de las Cosas

Como se ha comentado anteriormente en la introducción de esta memoria, es evidente que existe una deficiencia notable en la seguridad en el Internet de las

Cosas. Cualquier dispositivo conectado puede ser objeto de ciberataques que comprometan la privacidad de los usuarios o la seguridad del entorno y actualmente las *cosas* no están dotadas de mecanismos de seguridad suficientes como para poder mantenerse a salvo de ataques contra ellas.

El concepto de Internet de las Cosas ya supone un reto de seguridad en si mismo. En un entorno en los que todos los objetos están conectados entre ellos y comparten información a partir de la cual se toman decisiones sobre escenarios reales cualquier fallo, falta de información o información errónea podría tener consecuencias no deseables. Por ejemplo, si en el sistema de control de un cruce de calles de una ciudad se produce un error en la información que tiene que recibir un vehículo para evitar una colisión, este fallo podría costar la vida de varias personas.

La información en IoT, al igual que en Internet, ha estar protegida del acceso de terceros no autorizados y viajar cifrada para evitar que sea interceptada o modificada por atacantes. Si no se cumplen estas premisas [2] [5], el Internet de las Cosas estará abocado al fracaso. Estos problemas pueden cubrirse mediante la adaptación de protocolos de seguridad que se utilizan actualmente en Internet para IoT, dotando de servicios de seguridad a las cosas y garantizando la protección de la información tanto almacenada como transmitida y protegiendo a los dispositivos de ser manipulados por terceros no autorizados.

2.2.1. El problema de la identificación

Internet de las Cosas supone que una enorme cantidad de dispositivos estarán conectados a la red, lo que conlleva un reto para encontrar un estándar capaz identificar cada uno de los objetos. La identificación en IoT debe cumplir con varios requisitos [10]:

1. Disponer de un mecanismo de descubrimiento de objetos en la red a nivel local para poder encontrar de forma fácil dispositivos o servicios particulares.
2. Permitir la movilidad de dispositivos y objetos entre diferentes redes y entornos.
3. Ser dinámica y escalable, permitiendo que nuevos objetos y servicios se incorporen a un entorno cambiante y el mecanismo de descubrimiento permita la interacción entre objetos y servicios nuevos para los que los dispositivos no estén preconfigurados.
4. Simple y eficiente de forma que pueda ser implementado por dispositivos con capacidades computacionales limitadas.
5. Interoperable, permitiendo que dispositivos de diferentes fabricantes o con diferentes características y capacidades interactúen.
6. Flexible y extensible, que permita realizar cambios en los dispositivos y añadir nuevas propiedades y características sin perder continuidad y compatibilidad.
7. Fiable y seguro, de forma que la información no se vea comprometida.

2.3. Introducción a los sistemas de identidad

Se puede definir la *identidad* como el conjunto de características, atributos y rasgos que un individuo –o cosa– dice tener u otros dicen de él. Así, la identidad de una persona se puede definir por su nombre, su edad, su lugar de nacimiento, su dirección, su número de teléfono, sus direcciones de email, por dónde ha estudiado, dónde trabaja y cuál es su rol, qué servicios utiliza, cuáles son sus hobbies, etc. Un individuo puede tener más de una identidad que utilizar en situaciones concretas. Por ejemplo, una persona puede identificarse como ciudadano de una ciudad, como dueño de una casa o como estudiante de una universidad.

El control de acceso a servicios o recursos involucra diferentes procesos de seguridad [11]:

- *Autenticación*, que va acompañada de la identificación, es el proceso por el cual se verifica que una entidad se corresponde con la identidad que dice ser. Este proceso requiere de confidencialidad e integridad para que sea seguro.
- *Autorización*, para determinar si el sujeto tiene permiso para acceder al servicio. Al igual que la autenticación, este proceso requiere también de confidencialidad e integridad.
- *Aplicación de la política*, en el que se verifica que los procesos anteriores no atentan contra la política de acceso al recurso.
- *Responsabilidad*, en el que se deja constancia en los registros de los procesos anteriores.

Por otro lado, la política de control de acceso a estos recursos puede enfocarse de diferentes maneras:

- *Discretionary Access Control* (DAC). Se basa en listas de control de acceso controladas por el dueño que determinan un acceso binario (permitido/denegado).
- *Mandatory Access Control* (MAC). La organización establece una política de control de acceso basada en la sensibilidad del recurso en una escala de varios niveles.
- *Role Based Access Control* (RBAC). Se establecen etiquetas de rol, que se utilizan como política de control de acceso.
- *Trusted Based Access Control* (TBAC). La política de control de acceso se basa en la confianza en el sujeto.

En la Web, es habitual que un usuario se registre en diferentes servicios en los que el control de acceso está *centralizado*. El usuario realiza el proceso de autenticación e identificación mediante un usuario/contraseña que envía al servidor central, quien autoriza o deniega el acceso al cliente siguiendo unas políticas de acceso que están controladas por el dominio administrativo. El usuario tendrá

entonces un conjunto usuario/contraseña para cada servicio en el que se haya registrado y no podrá utilizar las credenciales en otros sitios.

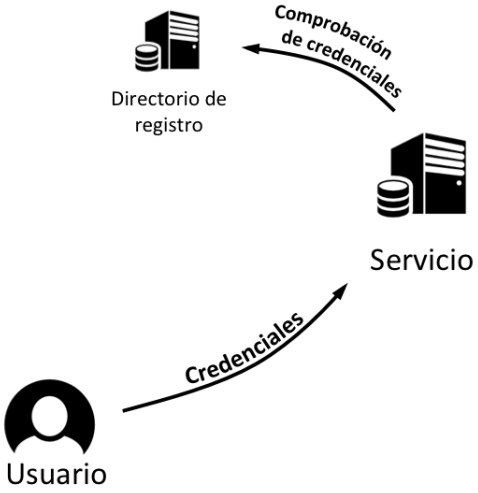

Figura 1: Control de acceso centralizado. Identificación por usuario/contraseña (Identidad 1.0)

En un sistema *distribuido* de control de acceso, existen diferentes fuentes de provisión de credenciales que el usuario puede utilizar para acceder a un servicio sin necesidad de que el usuario se hubiera registrado previamente. El usuario tiene libertad para autenticarse con diferentes credenciales según sus intereses. Este esquema de control de acceso rompe con los dominios administrativos de los sistemas de control de acceso centralizados, otorgando al usuario mayor libertad y flexibilidad.

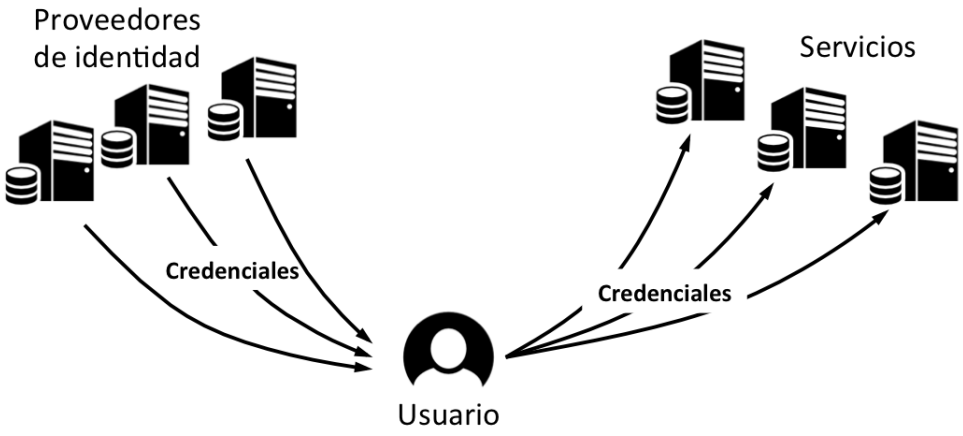

Figura 2: Control de acceso distribuido.

En un sistema de identidad digital el usuario se sitúa en el centro de la identificación y tiene el control total sobre sus datos, pudiendo elegir qué credencial

o credenciales proporcionar a cada servicio, lo que aporta flexibilidad al usuario y permite la interoperabilidad entre diferentes servicios. La *Identidad 2.0* permite diferentes mecanismos de autenticación, según los cuales otorga niveles de autenticación. Las credenciales y atributos de un usuario son emitidas por proveedores de su confianza, con los cuales tiene alguna relación. Estos proveedores almacenan la información del usuario y proporcionan al usuario el reconocimiento en otros lugares que confían en el proveedor lo que otorga a estos sistemas escalabilidad.

Sin embargo, para un número de proveedores de identidad muy elevado, no es posible que los servicios tengan certificados de todos los proveedores. La última generación de identidad digital, *Identidad 3.0* [12], incorpora la variable de riesgo, y toma decisiones de confianza en base a ello. Además, no solo se centra en la identidad y autenticidad, sino que también tiene en cuenta la privacidad del usuario proporcionando a los servicios solamente la información necesaria para su autorización.

2.4. OpenID

Figura 3: Logo de OpenID

OpenID es un protocolo de autenticación abierto y distribuido que permite a los usuarios autenticarse en servicios que admitan el estándar y proporcionar una identidad digital a través de proveedores de identidad terceros. Para identificarse, el usuario proporciona un identificador de tipo URI o XRI personal a su elección. Este identificador está asociado a un proveedor de identidad, a través del cual el usuario debe autenticarse utilizando los mecanismos que el proveedor requiera (usuario-contraseña, certificado digital, tarjeta de identidad, smartcard, biometría...). Una vez autenticado, el servicio al que el usuario desea acceder comprueba que la identidad del usuario es correcta a través del mismo proveedor de identidad.

El uso de OpenID como mecanismo de autenticación presenta como ventaja la posibilidad que un usuario se autentique en diferentes sitios o servicios diferentes e independientes con una misma identidad, proporcionada por su proveedor de identidad, sin necesidad de que el usuario proporcione al sitio directamente ningún tipo de información o se registre en cada uno de los servicios.

OpenID fue concebido para la Web. Actualmente la autenticación por OpenID está muy extendida gracias al dominio de proveedores como Google, Facebook, Twitter, Yahoo o LinkedIn, que son aceptados en muchos sitios web y aplicaciones que permiten el acceso con un solo click en el logotipo del proveedor correspondiente de forma fácil y cómoda para el usuario.

La OpenID Foundation (OIDF), una fundación sin ánimo de lucro, es la encargada de la estandarización del protocolo y de la promoción y protección de las tecnologías y comunidad de OpenID.

2.4.1. Explicación del protocolo

OpenID [13] utiliza mensajes de tipo Request/Response estándar de HTTP(S) como protocolo de transporte, de forma que sea compatible con cualquier navegador. En el proceso de autenticación, intervienen tres entidades:

- *End-user*: usuario final que desea autenticarse en un servicio y proporciona un identificador (*Identifier*) de tipo URI o XRI que asegura controlar (*Claimed Identifier*).
- *Relying Party* (*RP*): servicio al cual quiere acceder el usuario demostrando que controla el identificador proporcionado.
- *OpenID Provider* (*OP*) o *Identity Provider* (*IdP*): servidor que provee al usuario de identificador y en el cual confía el RP para que le asegure que el identificador proporcionado esta realmente controlado por el usuario.

El flujo de mensajes para la autenticación de un usuario en un Relying Party sigue el siguiente proceso:

1. El usuario accede al servicio iniciando el proceso de autenticación aportando un identificador al RP.
2. El RP normaliza el identificador aportado por el usuario y realiza un proceso de descubrimiento de la dirección del IdP.
3. (Opcional) Se establece una asociación entre el RP y el IdP mediante el intercambio de secreto compartido utilizando el protocolo de establecimiento de claves Diffie-Hellman. El IdP utilizará esta asociación para firma los siguientes mensajes y el RP para verificarlos.
4. Se redirecciona al usuario al IdP con un mensaje *Authentication request*.
5. El usuario se autentica ante el IdP y éste decide si el usuario está autorizado. La forma en la que el usuario se autentica ante el IdP no se especifica en el protocolo.
6. El IdP redirecciona al usuario al RP con un mensaje de *Authentication approved* o *Authentication failed*.

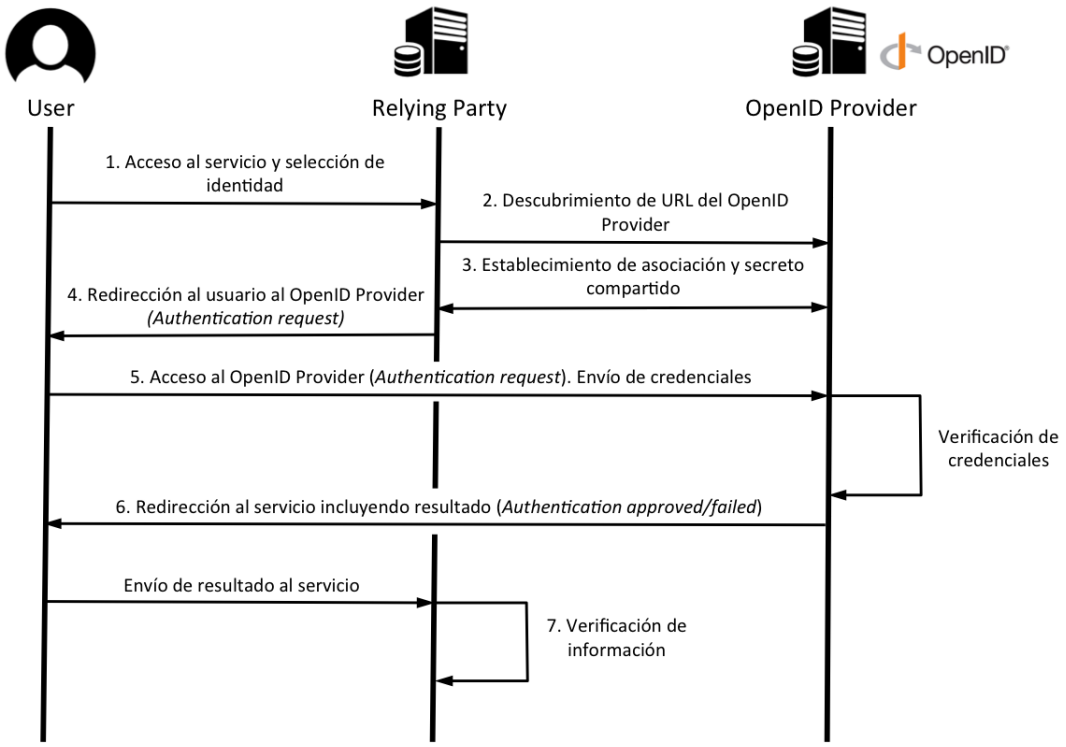

Figura 4: Flujo de mensajes OpenID

7. El RP verifica la información contenida en el resultado de la autenticación y comprueba la firma usando la clave compartida establecida en el paso 3.

Los mensajes enviados en OpenID vía HTTP son de tipo clave-valor. Cada mensaje, en función del tipo que sea, tiene unos parámetros que contienen la información del usuario y del proceso de autenticación. Por ejemplo:

```
openid.ns:http://specs.openid.net/auth/2.0
openid.mode:checkid_setup
```

Adicionalmente, se pueden utilizar extensiones para proveer de información extra sobre el usuario en el proceso de autenticación. Por ejemplo:

```
openid.ext1.type.attr1:name
openid.ext1.value.attr1:Wolfgang Amadeus
openid.ext1.type.attr2:surname
openid.ext1.value.attr2:Mozart
```

2.5. El modelo de actores

El *modelo de actores* es un modelo de computación propuesto en 1977 por Carl Hewitt. El modelo de actores está basado en entidades computacionales distribuidas, independientes, dinámicas y concurrentes que reciben el nombre de *actores* [14].

Cada actor tiene capacidad para comunicarse con otros actores mediante mensajes, aunque también tienen la posibilidad de poder comunicarse con entidades externas. Los actores tienen asociados una dirección y un buzón para recibir los mensajes de forma secuencial y procesarlos. Al procesar un mensaje un actor puede reaccionar:

- Cambiando su estado local.
- Enviando comunicaciones a otros actores.
- Creando nuevos actores.

Las comunicaciones entre actores son asíncronas, aunque también pueden ser síncronas, lo cual conlleva el bloqueo de los actores mientras se recibe la comunicación. En todo caso, el modelo de actores garantiza que los mensajes son entregados.

Las características de este modelo de computación hacen que presente muchas ventajas para la computación distribuida y el *cloud-computing*, al ser los actores independientes y poder estar distribuidos físicamente en el espacio. Además, la posibilidad de crear nuevos actores (o deshacerse de ellos) es ideal para servicios que requieran de flexibilidad y escalabilidad.

Los grande sistemas están compuestos de otros más pequeños. Las aplicaciones actuales requieren de *sistemas reactivos* que sean robustos, tolerantes a fallos, elásticos, flexibles, escalables, responsivos y orientados a mensajes [15]. Estas características pueden ser cubiertas por el modelo de actores para cubrir las necesidades de computación actuales.

> *"Actor approach is about the future of computing."*
>
> *Varol Akman*, in [16]

2.6. Akka

Figura 5: Logo de Akka

Akka es un *framework* o plataforma de código abierto para Scala y Java que implementa el modelo de actores ofreciendo un mayor nivel de abstracción para desarrollar aplicaciones escalables, flexibles y 'reactivas' –tal y como se define en [15]– y proporcionando transparencia ante la distribución de los actores [17]. Akka adopta un modelo de tolerancia de fallos denominado 'let-it-crash', muy utilizado en las telecomunicaciones, haciendo que las aplicaciones sean robustas.

La capacidad para escalar tanto en dimensión como en distribución hace que Akka sea una buena herramienta para numerosas aplicaciones que requieran comunicaciones asíncronas y concurrentes con un alto tráfico de datos y baja latencia. Actualmente, Akka está siendo utilizado en diferentes industrias:

- Telecomunicaciones
- Banca y finanzas
- Juegos on-line
- Servicios Backend (BaaS)
- Simulaciones
- *Business Intelligence* y *Data Mining*

2.6.1. Organización y estructura de los actores en Akka

En Akka, los actores son las unidades más pequeñas. Un actor ocupa aproximadamente 300 *bytes*. Cada actor tiene un estado, un comportamiento, un buzón de correo, actores hijos y un supervisor de estrategia, además de una referencia.

- *Referencia*: es una representación del actor cuyo principal propósito es poder enviar mensajes al actor al que se referencia.
- *Estado*: está representado por las variables que contiene un actor.
- *Comportamiento*: esta definido por la reacción de un actor al recibir un mensaje. El comportamiento puede variar con el tiempo o según el estado del actor.
- *Buzón*: es el contenedor en el que cada actor reciben los mensajes de otros actores para procesarlos. Los mensajes son parte fundamental de Akka, ya que dan comunicación a los actores. El buzón puede acumular múltiples mensajes, y la forma en que el actor se puede modificar; por defecto, el orden de procesamiento de los mensajes es FIFO (se procesa primero el mensaje que antes llegó).
- *Actores hijos*: un actor puede crear nuevos actores bajo su supervisión y delegarles subtareas. Así como los crea, también puede detenerlos.
- *Supervisor de estrategia*: es una parte del actor que se encarga de la supervisión del comportamiento de los actores hijos.

Los actores están organizados siguiendo una organización jerárquica. Cuando un actor recibe una tarea, puede dividirla en tareas más pequeñas y crear actores a los que delegar esa tarea bajo su supervisión, y hacer esto de forma recursiva. La clave del *sistema de actores* es reducir las tareas hasta que se puedan resolver directamente. Si un actor encuentra algún problema durante su ejecución, éste

envía un mensaje a su supervisor, que decidirá qué hacer (continuar, volver a empezar, parar o informar a su superior). Cada actor tiene, además de su referencia, una dirección que representa la jerarquía a la que pertenece.

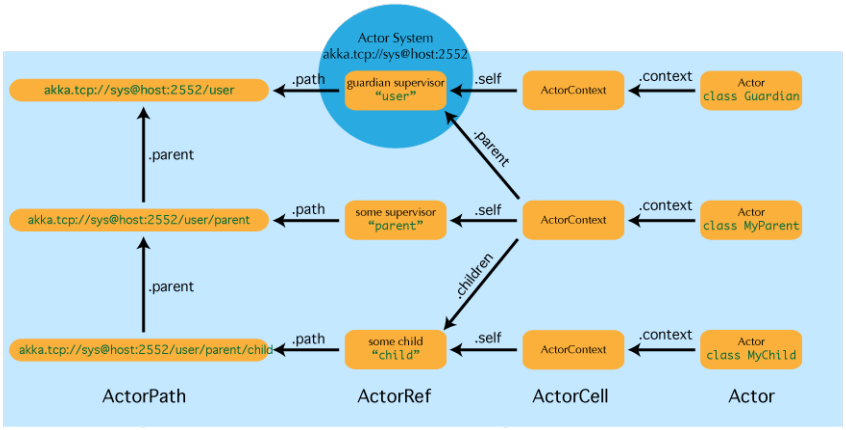

Figura 6: Relación entre actores dentro de un sistema de actores.

Este sistema de actores supervisados es transparente a la ubicación de los actores, por lo que la dirección de un actor puede ser local o remota, utilizando como protocolos de transporte TCP o UDP. Por ejemplo:

```
akka://actor-system/user/service/actor1              #local
akka.tcp://actor-system@host.com:1234/user/service   #remota
```

No debe confundirse la referencia del actor con la dirección. La dirección a un actor puede existir sin que exista un actor con esa dirección, mientras que la referencia a un actor no puede existir sin actor. También, una dirección puede estar asociada a un actor con referencia *'x'* y posteriormente a otro actor con referencia *'y'*. Por otro lado, los actores tienen una *dirección lógica*, que se obtiene por los enlaces de supervisión entre actores y representa la funcionalidad dentro del sistema de actores, y una *dirección física*, que representa el sistema de actores en el que el actor se está ejecutando. La dirección lógica y la física puede no coincidir, dada la naturaleza distribuida de Akka.

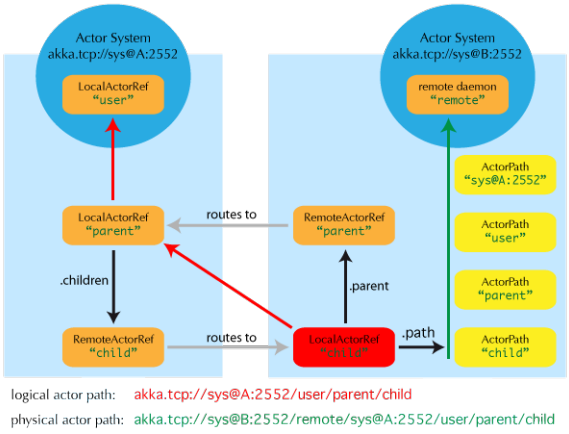

Figura 7: Diferencia entre dirección lógica y física de una actor dentro de un sistema de actores.

3. Diseño de un entorno de simulación para Internet de las Cosas

En estas sección se analizarán los requisitos necesarios para el desarrollo de un entorno de simulación para Internet de las Cosas, se analizarán las características que presentarán los escenarios de IoT y se elegirá el software sobre el cual se desarrollará el entorno.

3.1. Requisitos

Internet de las Cosas estará formado por una gran variedad de dispositivos de toda naturaleza que intercambiarán mensajes e información entre ellos de forma autónoma y concurrente. Para ello, todos los dispositivos y funciones deben tener capacidad de conexión y comunicación con el resto del entorno sin necesidad de intervención humana. Para lograr sistemas autónomos, es requisito necesario que los dispositivos dispongan de un mecanismo de descubrimiento a través del cual puedan entablar comunicación con otros dispositivos del entorno.

Las *cosas* formarán una red muy heterogénea tanto en la funcionalidad que tengan en IoT como en la capacidad de cómputo de cada uno de los dispositivos, que irán desde sencillos sensores que recolecten información de forma constante y cuya capacidad de cómputo esté muy limitada hasta grandes clústeres de servidores que recibirán y procesarán toda la información del entorno. Por tanto, es necesario poder modelar un entorno heterogéneo en funcionalidad y en capacidad de cómputo.

Por ello, el modelo de identidad digital para IoT debe ser sencillo para que funcione en cualquier dispositivo sin importar su capacidad de cómputo, pero a la vez debe ser compatible y exportable a otros modelos de identificación más complejos para los sistemas que lo requieran.

La topología de red que seguirá IoT no seguirá una estructura rígida y definida, por lo que ha de ser flexible y descentralizada de forma que cualquier dispositivo se pueda acoplar al entorno.

En resumen, los requisitos necesarios para el desarrollo de un entorno de simulación para Internet de las Cosas son:

- Concurrencia
- Conexión y comunicación
- Autonomía
- Autodescubrimiento
- Sencillez
- Exportable
- Topología no definida
- Descentralizado

3.2. Modelo de actores

En este apartado se explicarán las razones por las que el modelo de actores es el modelo de computación que mejor puede modelar un entorno de Internet de las Cosas.

3.2.1. Características de los actores para modelar un entorno concurrente, heterogéneo y sin topología definida

Aunque el Internet de las Cosas todavía no ha alcanzado la madurez suficiente como para poder definirlo y modelarlo con exactitud, hay ciertas características que sí podemos considerar evidentes: concurrencia, heterogeneidad y topología indefinida.

Internet de las Cosas estará formado por cientos, miles, millones o billones de *cosas*, dependiendo de la escala a la que lo consideremos, y estarán funcionando continuamente de forma concurrente. La principal característica del modelo de actores [14] es la concurrencia en su ejecución, por lo que permiten modelar un entorno en el que cada cosa esta representada por un actor. El modelo de actores se basa en la comunicación entre actores por mensajes asíncronos, lo que se corresponde con fidelidad al comportamiento que tendrían las cosas en un entorno real.

La escalabilidad que ofrece el modelo de actores permite simular escenarios de diferentes magnitudes, desde escenarios pequeños como una casa con decenas de dispositivos hasta un entorno de Smart City con millones de ellos. Además, la flexibilidad en la estructura de los sistemas de actores da pie a una topología libre que se adapte a cada escenario.

Por otro lado, la libertad a la hora de definir el comportamiento de un actor posibilita poder simular cualquier tipo de dispositivo. Asimismo, cada actor tiene la posibilidad de crear nuevos actores hijos bajo su supervisión, lo que ofrece poder simular diferentes funcionalidades dentro del dispositivo o nuevos dispositivos diferentes bajo su mando.

Estas características del modelo de actores, hace que sea la herramienta adecuada para desarrollar un entorno de simulación de escenarios de Internet de las Cosas.

3.2.2. Características esperables de los entornos de IoT que hacen interesante el uso de actores

El desarrollo del Internet de las Cosas se encuentra aún en una fase prematura de crecimiento. Sin embargo, caben esperar ciertas características que refuerzan la idoneidad de los actores aplicados a entornos IoT.

Las *cosas* formarán una red de grandes dimensiones y dependerán unas de otras para el intercambio de información. Esta información se transmitirá en mensajes,

pero se puede esperar que las cosas sean independientes entre si y su funcionamiento esté débilmente acoplado al de los demás dispositivos del entorno, de modo que el fallo de un dispositivo no ocasione el fallo de otros.

Cabe esperar también que los sistemas sean *responsivos*, es decir, que ante cualquier problema el sistema responda de forma adecuada detectando y tratando el problema de forma efectiva y rápida. Esto tiene que ver con la *resiliencia*, es decir, la capacidad del sistema para recuperarse después de tratar y superar el error, que se consigue mediante la contención y aislamiento del error y la réplica y delegación de las partes del sistema afectadas, consiguiendo así un sistema robusto. En relación con la responsividad, los sistemas deberán también reaccionar ante el aumento o disminución de peticiones o usuarios en un entorno, siendo *elásticos* y escalables para poder satisfacer la demanda de forma eficiente [15].

Por otro lado, otra propiedad que con seguridad caracterizará IoT será la movilidad de las cosas por diferentes entornos, lo que encaja con la transparencia de ubicación de los actores y la posibilidad de que un actor se mueva o se replique en diferentes ubicaciones.

Otra de las características que se pueden esperar en los entornos de IoT es, por un lado, que las *cosas* tengan más de una aplicación o funcionalidad. Por ejemplo, utilizando el ejemplo del termostato que se usó previamente en el apartado 1.3, un termostato puede tener una aplicación de usuario para sus preferencias de temperatura, una aplicación del servicio de mantenimiento que envía alertas cuando algo no va bien y una aplicación de la compañía energética para estimar la demanda de energía. Cada una de ellas informa y mantiene comunicación con puntos diferentes, por lo que el modelo de actores de actores es adecuado para IoT. Por otro lado, y continuando con el ejemplo, a la aplicación de usuario podrían acceder diferentes personas, como el propietario, sus hijos o sus invitados, y cada uno de ellos tendrá unos permisos específicos para el acceso a la aplicación de usuario, es decir, cada uno de ellos tendrá un rol diferente.

3.3. Uso de identidad en IoT

El uso de identidad digital en entornos de Internet de las Cosas es necesario para garantizar la seguridad en el control de acceso a dispositivos, recursos, aplicaciones e información y poder diferenciar entre accesos autorizados y no autorizados. Existen diferentes protocolos que implementan la tecnología de identidad digital que pueden reutilizarse para su integración en IoT.

Para que el uso de identidad digital se integre con éxito en IoT en su eminente expansión, es necesaria la utilización de protocolos abiertos que permitan la estandarización para que cualquier dispositivo sea capaz utilizar la identidad digital con el resto, evitando la proliferación de entornos cerrados y silos verticales incompatibles entre ellos.

Dada la gran variedad de dispositivos y servicios que pueden surgir, debe de ser posible el intercambio de atributos de forma libre y flexible que permita que el mecanismo de identidad digital pueda adaptarse a cualquier entorno. Además, debe poder ser implementado por los dispositivos con menor capacidad de cómputo y compatible y exportable a otros sistemas de identificación más sofisticados.

3.4. Elección de software

En estas sección se explicarán los motivos de elección del software a utilizar en el desarrollo del entorno de simulación para Internet de las Cosas.

3.4.1. Akka

De entre todas las implementaciones del modelo de actores, se ha elegido el *framework* Akka para este proyecto. Las razones que han llevado a la elección de Akka son las siguientes:

1. Akka es código abierto bajo licencia Apache 2.
2. Akka cuenta con una API para Java, que es un lenguaje de programación muy habitual y con el que el autor de este proyecto se siente más cómodo trabajando.
3. Cuenta con un una documentación muy completa y bien explicada, que facilita la comprensión tanto del modelo de actores como del uso de la plataforma.
4. Es muy fácil de usar.
5. Cuenta con una amplia comunidad en la red.
6. Es un modelo de actores muy bien implementado.
7. Akka cada vez se utiliza más en proyectos, por lo que tener conocimientos de ello es un punto positivo a tener en cuenta.

3.4.2. OpenID

La tecnología de identidad digital elegida para este proyecto es OpenID. Las razones por las que se ha elegido este protocolo son las siguientes:

1. OpenID es un estándar abierto.
2. Es un protocolo conocido y extendido en la Web.
3. OpenID es un protocolo de identificación sencillo y suficiente para los propósitos de este proyecto.
4. Es una solución ligera, por lo que los dispositivos con mayores limitaciones no tendrán problema en utilizarlo.

En este proyecto se han utilizado las librerías de *openid4java*, que es una implementación en Java del protocolo OpenID bajo licencia Apache 2 [18].

3.5. Arquitectura

En esta sección se definirá la arquitectura de la plataforma de simulación y se explicarán los elementos que la formarán. Para ello, se recurrirá al siguiente ejemplo:

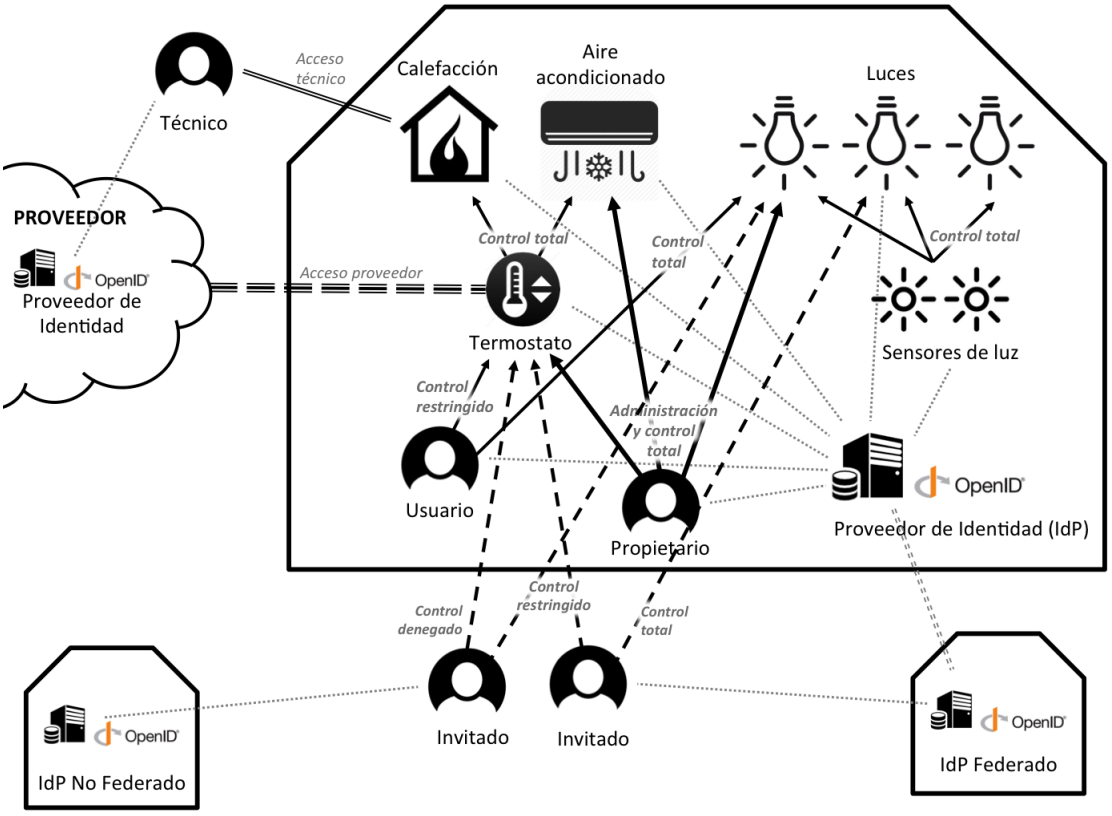

Figura 8: Esquema de arquitectura.

En el ejemplo se muestra una *Smart Home* del año 2020, en la que hay un sistema de alumbrado inteligente controlado por sensores de luz y un sistema de climatización, formado por calefacción y aire acondicionado, controlado por un termostato, controlado por un termostato inteligente. En la casa, vive una familia compuesta por el propietario de la casa y otros usuarios. Además, como sistema de control de acceso a los elementos conectados de la casa, hay un Proveedor de Identidad (IdP) basado en tecnología OpenID.

Habitualmente, a la casa acuden amigos y familiares de confianza, quienes poseen también de una *Smart Home* igual que la descrita. Ocasionalmente, también acuden otros amigos y familiares de menos confianza, también con *Smart Home*.

El propietario de la casa tiene un contrato con un proveedor de energía por el cual el proveedor tiene acceso al termostato para recopilar datos de los hábitos de consumo de la familia para mejorar la eficiencia energética global a cambio de un servicio técnico telemático que resolverá cualquier incidencia con la calefacción y

aire acondicionado de forma remota.

El propietario tiene acceso de administración y control pleno a todos los dispositivos de la casa, y ha decidido que el resto de su familia puede controlar libremente el sistema de alumbrado, pero el sistema de climatización solo de forma restringida, cambiando en 3°C arriba o abajo la temperatura fijada en el termostato. De igual forma, sus invitados de confianza tiene control total sobre el alumbrado, pero solo tendrán acceso a consultar la temperatura del termostato, no a modificarla. Sus invitados de menor confianza podrán controlar el alumbrado de ciertas zonas de la casa, pero no tendrán acceso ninguno al sistema de climatización. Los sensores de luz controlarán la programación de alumbrado establecida por el propietario, siempre que no se diga lo contrario.

De este escenario de ejemplo se pueden extraer cuales son los elementos fundamentales de la arquitectura:

- Proveedor de Identidad (IdP)
- Servicio o Relying Party (RP)
- Usuarios y actuadores
- Sensores

Cabe destacar el papel que toma el termostato en el ejemplo: actúa como servicio al que acceden los usuarios, como actuador al controlar el funcionamiento de la calefacción y aire acondicionado y como sensor al controlar la temperatura del entorno.

A continuación se explica cual es la función de cada uno, cómo funcionan y cómo se comportan.

3.5.1. Usuarios

Se define *usuario* como una entidad humana capaz de interactuar con los elementos de la red a la que se encuentra conectado y que hace uso de los servicios que estos proporcionan.

Los usuarios pueden disponer de múltiples identidades, cada una emitida por un proveedor de identidad. Estas identidades puede utilizarlas según el contexto y dominio administrativo en el que el usuario quiera actuar. Por ejemplo, el propietario de la casa podrá utilizar la identidad del IdP de su casa para interactuar dentro de su casa, pero en casa de otra persona esa identidad (en principio) no tendría validez. De igual forma, esta identidad podría no tener validez (completa o parcial) para identificarse en una *Smart City*.

Una identidad de un usuario puede estar asociada a diferentes roles. Por ejemplo, el propietario de la casa del ejemplo puede actuar con el rol de usuario propietario, teniendo acceso al control de los dispositivos, o con el rol de administrador,

teniendo acceso a la configuración de los dispositivos.

Los usuarios pueden comunicarse con los dispositivos del entorno que les rodea para interactuar con ellos. Esto puede requerir un proceso de identificación, en cuyo caso el usuario debe autenticar su identidad aportando la aprobación del IdP.

Mediante el intercambio de atributos que dan forma a las diferentes identidades de cada usurario en función del servicio al que deseen acceder, se aumenta la seguridad tanto de los usuarios, preservando lo máximo posible su privacidad, como de los servicios, que pueden requerir un conjunto de atributos para que un usuario pueda acceder.

En en el entorno de simulación de Akka, los usuarios son actores que pueden enviar mensajes a otros actores y se comunican con el IdP para autenticar la identidad que proporcionen a los servicios.

3.5.2. Actuadores

Se define *actuador* como un dispositivo o *cosa* con capacidad para tomar decisiones de forma autónoma y controlar o enviar órdenes a otros dispositivos o *cosas*.

Un actuador dispone de una identidad emitida por el proveedor de identidad de su dominio administrativo, que utiliza para enviar mensajes a otras cosas con órdenes. Los actuadores pueden enviar mensajes y órdenes dentro de su dominio administrativo.

Al igual que los usuarios, los actuadores hacen uso del envío de atributos que definen y modelan su identidad para el acceso a los diferentes servicios de su entorno, en función de lo que cada servicio requiera.

En el entorno de simulación de Akka, los actuadores son actores que se comunican con otros actores que representen servicios, y se comunican con el IdP para autenticar su identidad ante los servicios.

3.5.3. Sensores

Los *sensores* son dispositivos que tienen la capacidad de medir variables del entorno en el que se encuentran para posteriormente proporcionarla a modo de información a usuarios, actuadores o servicios.

Los sensores son parte fundamental del concepto de Internet de las Cosas, ya que dotan a los sistemas de la capacidad de tomar información del mundo físico para tomar decisiones en base a ello.

A la hora de identificarse, un sensor puede detallar sus capacidades mediante el intercambio de diferentes atributos que definan sus posibilidades. Un mismo

dispositivo puede disponer de diferentes sensores para realizar diferentes tipos de mediciones del entorno y, a la hora de identificarse ante un servicio, puede hacer uso de uno o varios atributos para dar a conocer sus capacidades como entidad 'sensor'.

Los sensores pueden comunicarse con usuarios, para proporcionarles información en tiempo real sobre variables del entorno físico, con actuadores, para proporcionarles información sobre la cual tomar decisiones autónomas, o con servicios, para que traten la información aportada como crean conveniente, ya sea almacenándola, procesándola, enviándola a otros dispositivos, etc.

3.5.4. Relying Party (RP)

El *Relying Party* es el servicio al que usuarios y actuadores acceden de forma segura después de autenticarse demostrando su identidad a través de un IdP de confianza.

El RP puede recibir mensajes tanto de los usuarios y como de los actuadores. Al recibir un mensaje para hacer uso del servicio, en función de la política de control de acceso que se haya establecido para ese servicio, el RP requerirá al usuario o actuador una identidad, que posteriormente deberá ser autenticada por un IdP de confianza. Mediante el intercambio de atributos, un RP puede solicitar a un usuario o actuador que aporte una serie de atributos, bien de forma obligatoria u opcional, para acceder al servicio que solicite.

Un mismo dispositivo puede tener varios servicios accesibles. La respuesta del RP a los mensajes que reciba puede ser diferente en función de la identidad o rol de cada usuario, pudiendo discriminar el acceso a los diferentes servicios en función de la identidad proporcionada.

En Akka, cada servicio será simulado como un actor, por lo que los dispositivos con servicios se simularán con un actor por cada servicio. El termostato del ejemplo, que contaría con dos servicios, uno para la climatización y otro para enviar los datos de consumo al proveedor energético.

El RP cuenta con un módulo de descubrimiento que utiliza para conocer la URI del IdP que emitió la identidad que un usuario ha utilizado y poder establecer una asociación segura mediante secreto compartido para verificar la identidad proporcionada por el usuario.

3.5.5. Proveedor de Identidad (IdP)

El *Proveedor de Identidad* es el elemento de la red encargado de autenticar a los usuarios y dispositivos a los que haya emitido una identidad y que deseen acceder a servicios que requieran de identificación. Además, es responsabilidad del IdP verificar para su autenticación los atributos requeridos y aportados en la identidad

por el dispositivo que desee acceder a un servicio.

El IdP puede mantener comunicación tanto con los usuarios y actuadores para autenticar su identidad como con los RP que solicitan el establecimiento de una asociación para verificar la identidad de los usuarios.

El IdP puede estar dentro del dominio administrativo o fuera, en cuyo caso necesitará estar federado para tener autoridad para la verificación de identidades. Tomando el ejemplo descrito anteriormente, la casa dispone de un IdP con autoridad sobre los dispositivos de la casa, su dominio administrativo. Por otro lado, los amigos y familiares de confianza que tienen ciertos permisos para interactuar utilizarán su IdP federado, que les proporcionará el servicio de identificación válido dentro del dominio administrativo del IdP de la casa. Sin embargo, los amigos y familiares de menor confianza, pese a tener también un IdP, éste no es federado y por tanto no tendrá la autoridad para proporcionar la verificación de identidad en la que puedan confiar los dispositivos de la casa.

4. Desarrollo de un entorno de simulación

En esta sección se explicará el proceso llevado a cabo para el desarrollo del entorno de simulación para Internet de las Cosas.

4.1. Modelado de cosas con actores

Para poder modelar una *cosa* con capacidad de conexión y comunicación a través del envío de mensajes, lo primero es disponer de una referencia a ella. Esta referencia es la dirección publica del actor de Akka, y será del tipo:

```
akka://actor-iot-system/thingname
```

Esta URI puede ser usada para enviar mensajes a la entidad a la que haga referencia, y todos los actores tendrán su propia dirección. Además, en el entorno de simulación, cada actor utilizará esta URI como identidad propia.

El concepto de *cosa* dentro de Internet de las Cosas es muy amplio. En el Internet de las Cosas cabe esperar que todo esté conectado, por ejemplo:

- Smartphone
- Smartwatch
- Smart TV
- Equipos de música
- Bombillas
- Enchufes
- Nevera
- Lavadora
- Robot de cocina
- Termostato
- Aire acondicionado
- Calefacción
- Cámaras de vigilancia
- Sensores de temperatura
- Sensores de luz
- Sensores de presencia
- Sensores de humedad
- Sensores de lluvia
- Contadores
- Elementos de distribución de red
- Centrales eléctricas
- Oleoductos y gaseoductos
- Vehículos
- Semáforos
- Señales
- Farolas
- Sensores de aparcamiento
- Sistemas de regulación de tráfico
- Servicios de emergencia
- Vehículos de emergencia
- Sensores de contaminación
- Sensores de visibilidad
- Estaciones de carga de vehículos
- ...

De todos estas *cosas* o sistemas, unas envían mensajes, otras reciben mensajes y otras envían y reciben. Unas *cosas* reaccionan a la recepción de un mensaje respondiendo con otro mensaje, otras reaccionan enviando un mensaje a una tercera *cosa* y otras simplemente guardan la información contenida en el mensaje. El comportamiento de cada dispositivo está fuera del objetivo de este proyecto, por lo que los mensajes que se intercambian están dirigidos al proceso de identificación.

Lo que tienen en común todas las *cosas* es que poseen una serie de atributos como un nombre, una localización física, una serie de capacidades y atributos que describan qué es y qué propiedades tiene. Estos atributos están relacionados con la finalidad de cada dispositivo, y la clasificación de las cosas y sus capacidades debería ser suficientemente flexibilidad y abierta para cumplir con los requisitos y objetivos globales para IoT expuestos anteriormente, pero no es el objetivo fundamental de este proyecto. Por tanto, para poder continuar con el desarrollo del entorno de simulación, se considerará un conjunto reducido y cerrado de atributos sencillos que caracterizan a las *cosas* para su identificación.

Además, las *cosas* también conocen la URI de su proveedor de identidad.

En resumen, las *cosas* en el entorno de simulación tienen los siguientes atributos:

- Identidad (URI)
- Dirección IdP
- Propiedades de la entidad
- Capacidades

4.2. Integración de OpenID en un sistema IoT

Las funciones que tienen los elementos de OpenID son asumidas por *cosas* en un sistema IoT. Así, los elementos se corresponden de la siguiente manera:

- Los usuario de OpenID son también los usuarios en IoT, además de *cosas* que utilicen servicios: actuadores.
- Los RP de OpenID son los servicios que ofrece cada *cosa*.
- El IdP de OpenID sigue siendo el IdP en IoT sin ningún cambio.

Para simplificar la simulación, el IdP admite que las *cosas* y usuarios se registren de forma sencilla mediante el envío de un mensaje *RegisterUser* de registro, de forma que el IdP actuará de forma favorable en el proceso de autenticación de OpenID con los identificadores de usuario que tenga registrados.

Cuando un actor que representa un servicio o RP recibe un mensaje de tipo *ActionRequest* por parte de un usuario o actuador, en el que se especifica una acción a realizar, el RP espera que incluya un *AuthSuccess* emitido por el IdP. Si no

es así, se inicia el proceso de identificación del usuario que ha realizado la petición siguiendo el proceso explicado en la sección 2.4.1 y realizando el intercambio de mensajes de la tabla 4.

```
ActionRequest request = new ActionRequest(action.getActuator().path().toString(),
        action.getAction(), getSelf().path().toString());
action.getActuator().tell(request, getSelf());
```

En primer lugar, el RP inicia un proceso de descubrimiento del IdP. OpenID utiliza el protocolo de descubrimiento Yadis para identificadores de tipo URL y XRI. Como solución simple para el desarrollo del entorno de simulación, el procedimiento de descubrimiento del IdP en un sistema de actores Akka se basa en que el propio usuario conozca la URI de su IdP. Para que el RP la conozca y pueda establecer contacto con el IdP, envía un mensaje de descubrimiento *DiscoverIdProvider* al usuario y espera la respuesta de éste con la URI de su IdP. La respuesta al proceso global de descubrimiento es un objeto *List* con los IdPs descubiertos.

```
List discoveries = manager.discover(req.getUserSupliedId(), getContext().system());
```

Una vez el RP tiene los resultados del descubrimiento intenta establecer una asociación con un IdP de la lista de los resultado. El establecimiento de la asociación segura utilizando el intercambio de claves Diffie-Hellman está ya implementado en las librerías *openid4java*.

```
DiscoveryInformation discovered = manager.associate(discoveries);
```

Un mensaje *AssociationRequest* enviado por un RP hacia un IdP tiene la siguiente forma:

```
openid.ns:http://specs.openid.net/auth/2.0
openid.mode:associate
openid.session_type:DH-SHA256
openid.assoc_type:HMAC-SHA256
openid.dh_consumer_public:AKnljwa+4alZ93RVSv2nAvr0SnZ3j5/JGkylwDwlCUiqdoP6tzs95NLn2vGkjHI
        qLli12MNUHsWhDRyJnEeo7r7oj4JOgA3MS9o1n8S5tFRWsbWV6XrkBdh9aE4V0YKzCKtdgDw8dP7AI5qr
        5vVHpl4s5noveYW5fu9gG1kW707N
```

El mensaje *AssociationResponse* enviado por el IdP al RP en respuesta al *AssociationRequest* tiene la siguiente forma:

```
ns:http://specs.openid.net/auth/2.0
session_type:DH-SHA256
assoc_type:HMAC-SHA256
assoc_handle:1442666930356-0
expires_in:1800
dh_server_public:U4qkjz+O2M/PYiy3TfxJmUO3/yi3nk2ybMBUcj4NH22KHkRADwu97quilmlYDnkGR+ktUPCP
        ktfl69dmLRzsheFykL6FPhOiantH/7YS9M1xu3pWLxMfr56CUsTfQ8dUXD+QXJx2EsT6tArFt/VvuFtpz
        un2XYOb0y5UsrfbfRo=
enc_mac_key:8K7rHeQICAlSUi11jo0XEavrSiS5RIE3qzY0lB8BbZs=
```

Con la asociación segura establecida con el IdP, se redirecciona al usuario al IdP con un mensaje *AuthRequest* para que se autentique. Adicionalmente se añade una extensión con los atributos requeridos para aceptar la autenticación.

```
AuthRequest authReq = manager.authenticate(discovered, getSelf().path().toString());
FetchRequest fetch = Capabilities.getAttributesRequired(action);
authReq.addExtension(fetch);
```

Un mensaje *AuthRequest* enviado por un RP `service-id` a un usuario `user-id` requiriendo que se autentique y proporcione los atributos `1`, `2` y `3` obligatoriamente y los atributos `4` y `5` de forma opcional tendría la siguiente forma:

```
openid.ns:http://specs.openid.net/auth/2.0
openid.claimed_id:akka://actor-iot-system/user/user-id
openid.identity:akka://actor-iot-system/user/user-id
openid.return_to:akka://actor-iot-system/user/service-id
openid.realm:akka://actor-iot-system/user/service-id
openid.assoc_handle:1442662730579-0
openid.mode:checkid_setup
openid.ns.ext1:http://openid.net/srv/ax/1.0
openid.ext1.mode:fetch_request
openid.ext1.type.attr1:akka://iot/thing/name
openid.ext1.type.attr2:akka://iot/thing/owner
openid.ext1.type.attr3:akka://iot/thing/location
openid.ext1.required:attr1,attr2,attr3
openid.ext1.type.attr4:akka://iot/thing/battery
openid.ext1.type.attr5:akka://iot/thing/mobility
openid.ext1.if_available:attr4,attr5
```

Este mensaje lo recibe el usuario y se lo reenvía al IdP. Tras un proceso en el que el usuario debe autenticarse ante el IdP mediante los mecanismos oportunos, que no se abarcan en este proyecto, el IdP le responde con un *AuthSuccess* en el que se incluye la identificación, el RP para el que es, parámetros relacionados con la asociación RP-IdP, para que el RP pueda verificar la autenticidad de la autenticación, los parámetros requeridos en el *AuthRequest* y la firma de los campos que se indican para asegurar la integridad del mensaje.

```
openid.ns:http://specs.openid.net/auth/2.0
openid.op_endpoint:akka://actor-iot-system/user/IdP
openid.claimed_id:akka://actor-iot-system/user/user-id
openid.response_nonce:2020-01-13T13:35:52Z0
openid.mode:id_res
openid.identity:akka://actor-iot-system/user/user-id
openid.return_to:akka://actor-iot-system/user/service-id
openid.assoc_handle:1442669752284-0
openid.signed:op_endpoint,claimed_id,identity,return_to,response_nonce,assoc_handle,ns.ext1,
        ext1.mode,ext1.type.attr1,ext1.value.attr1,ext1.type.attr2,ext1.value.attr2,
        ext1.type.attr3,ext1.value.attr3,ext1.type.attr4,ext1.value.attr4,ext1.type.attr5,
        ext1.value.attr5,ext1.type.attr6,ext1.value.attr6,ext1.type.attr7,ext1.value.attr7
openid.sig:g7PtPWSI7q78298+fS43BoTOekOm7wf6Xi++yIK2DhE=
openid.ns.ext1:http://openid.net/srv/ax/1.0
openid.ext1.mode:fetch_response
openid.ext1.type.attr1:name
openid.ext1.value.attr1:client.ClientActor@359ba2ce
openid.ext1.type.attr2:owner
openid.ext1.value.attr2:akka://actor-iot-system
openid.ext1.type.attr3:location
openid.ext1.value.attr3:home
openid.ext1.type.attr4:battery
openid.ext1.value.attr4:yes
openid.ext1.type.attr5:mobility
openid.ext1.value.attr5:yes
```

Al recibir el usuario el *AuthSuccess*, vuelve a enviar el mensaje *ActionRequest* incluyendo el *AuthSuccess*:

```
ActionRequest action = new ActionRequest(act.getService(), act.getAction(),
                                                act.getUserSupliedId(), authSuccess);
context().actorSelection(act.getService()).tell(action, getSelf());
```

4.3. Integración de Akka como protocolo de transporte para OpenID

El protocolo de OpenID fue diseñado para la web utilizando como protocolo de transporte HTTP y HTTPs, y así están implementadas las librerías de *openid4java* utilizadas. Aunque la idea de utilizar como protocolo HTTP puede ser viable de cara al acceso vía web en un entorno real, se ha integrado el protocolo de transporte de Akka para utilizarlo en el intercambio de mensajes del protocolo en el entorno de simulación. Para ello, se han modificado las partes de las librerías en las que se hace uso de URLs que requieren como protocolo de transporte http o https. A continuación se exponen los módulos más importantes de la librería que se han modificado para la integración de Akka.

El método `public static URL normalize(String text, boolean removeFragment)` de la clase `UrlIdentifier` se utiliza para normalizar los identificadores de tipo URI y XRI que demande el usuario. Los métodos de normalización utilizados en las librerías no admiten como protocolo `akka`, por lo que se han modificado para que sean admitidos. Así, el método `normalize` devuelve una URL normalizada cuyo protocolo de transporte es `akka`.

```
if(uri.getScheme().equals("akka"))
    url = new URL(null, text , new Handler());
else
    url = uri.normalize().toURL();

String protocol = url.getProtocol().toLowerCase();
String host = url.getHost().toLowerCase();
int port = url.getPort();
String path = normalizeUrlEncoding(url.getPath());
String query = normalizeUrlEncoding(url.getQuery());
String fragment = normalizeUrlEncoding(url.getRef());

[...]

URL normalized;
if(!protocol.equals("akka"))
    normalized = new URL(protocol, host, port, file);
else
    normalized = url;

return normalized;
```

Esta clase se utiliza posteriormente para trabajar con URIs, por lo que, una vez modificada la clase para que admita el protocolo de transporte de Akka, se puede utilizar libremente.

A continuación se modifican las partes del código que se utilizan según el orden que sigue el intercambio de mensajes del protocolo OpenID. Así, el siguiente módulo a modificar es el de descubrimiento, contenido en la clase `Discovery`. El método `public List discover(Identifier identifier)` es el que realiza el procedimiento de descubrimiento del IdP, y está implementado para utilizar identificadores de tipo XRI o URL, en cuyo caso utiliza el protocolo Yadis. Para que el procedimiento de descubrimiento funcione como se ha explicado anteriormente con el protocolo `akka`, se introducen las siguientes modificaciones en el código:

```
if (identifier instanceof XriIdentifier){ [...] }
```

```
else if(identifier instanceof UrlIdentifier &&
                ((UrlIdentifier)identifier).getUrl().getProtocol().equalsIgnoreCase("akka")){
    UrlIdentifier urlId = (UrlIdentifier) identifier;
    result = _akkaResolver.discover(urlId, _actorSystem);
}

else if (identifier instanceof UrlIdentifier){ [...] }

else{
    throw new DiscoveryException("Unknown identifier type: " + identifier.toString());
}
return result;
```

Cuando se obtiene el resultado del descubrimiento, el siguiente paso es que el RP establezca una asociación con el IdP. Para ello, deben realizar un intercambio de mensajes, por lo que es necesario utilizar un protocolo de transporte. Este intercambio de mensajes tiene lugar en la clase `ConsumerManager` en el método `private int call(String url, Message request, ParameterList response, ActorSystem system)` , que es llamado desde `private int associate(DiscoveryInformation discovered, int maxAttempts)` . Dentro del método `call`, se introduce el siguiente `if-else` para que, si el protocolo del identificador es `akka`, utilice la mensajería de Akka para establecer la asociación:

```
if (urlIdentifier.getUrl().getProtocol().toString().equalsIgnoreCase("akka")){
    try{
        AkkaFetcher fetcher = new AkkaFetcher(urlIdentifier, system);
        Message resp = (Message) fetcher.ask(request);
        if(resp instanceof AssociationResponse)
            responseCode = HttpStatus.SC_OK;
        else if(resp instanceof AssociationError)
            responseCode = HttpStatus.SC_BAD_REQUEST;
        response.copyOf(resp.getParams());
        } catch(Exception e){
            return HttpStatus.SC_BAD_REQUEST;
        }
}
else{ [...] }
return responseCode;
```

Como se aprecia en el código, aunque se utilice como transporte Akka, se utilizan los códigos de estado de HTTP.

En el otro extremo, el IdP recibe el mensaje por Akka y realiza el procesamiento del mensaje OpenID de la forma habitual [19]. La respuesta es posteriormente enviada por el transporte de Akka:

```
private void processAssociationRequest(AssociationRequest message, ActorRef sender)
        throws Exception{

    [...]

    sender.tell(response, getSelf());
}
```

5. Escenarios y pruebas

En esta sección se plantearán posibles escenarios reales de uso y se explicarán los resultados de las pruebas llevadas a cabo sobre ellos.

5.1. Escenarios de uso

Pese a que Internet de las Cosas es un concepto general que lo engloba todo, la realidad es que actualmente existe una división en silos aislados. A continuación se plantean diferentes escenarios sobre los cuales este proyecto puede aportar una mayor seguridad.

5.1.1. Escenario genérico

Los escenarios de uso reales que pueden proponerse pueden ir desde escenarios muy simples en el que intervienen dos *cosas* hasta enormes ecosistemas que abarquen diferentes ámbitos reales unidos entre si para intercambiar información que pueda resultar de utilidad entre ellos. Sin embargo, todos los escenarios siguen una estructura común que se componen de cuatro tipos de elementos [ver 3.5]:

- Usuario/actuador
- Sensor
- Servicio (RP)
- Proveedor de Identidad (IdP)

Es indispensable disponer de un servicio y, para el propósito del proyecto, un IdP. Usuarios, actuadores y sensores son los elementos que aportarán el dinamismo al entorno, y puede haber usuarios, actuadores, sensores o una mezcla de ellos. Estos se comunicarán con el servicio para aportar/solicitar información o dar una orden, para lo cual tendrían que realizar el proceso de identificación correspondiente:

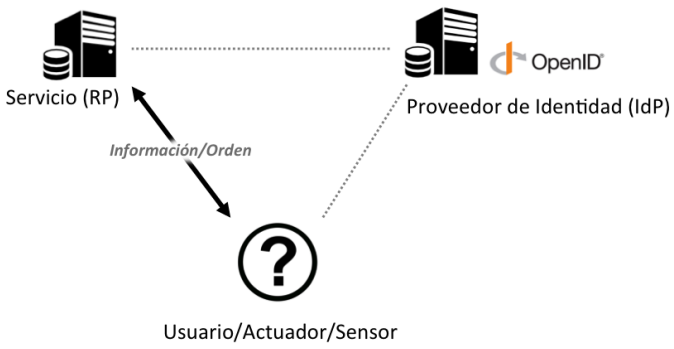

Figura 9: Escenario IoT genérico

5.1.2. Escenario: *Smart Home*

El escenario de una *Smart Home* es el que más se ha utilizado a lo largo de esta memoria. Dentro de una *Smart Home* existen numerosos escenarios diferentes sobre los que el uso de IoT sería beneficioso para el usuario. A continuación se explica una situación diferente al ejemplo que se a utilizado en secciones previas.

Supóngase una *Smart Home* en la que hay instalada una alarma de seguridad que el propietario activa y desactiva de forma remota. En la casa hay una serie de sensores de presencia instalados, que sirven para detectar movimiento. Además, los sensores de movimiento se utilizan para optimizar el uso de energía dentro de la casa.

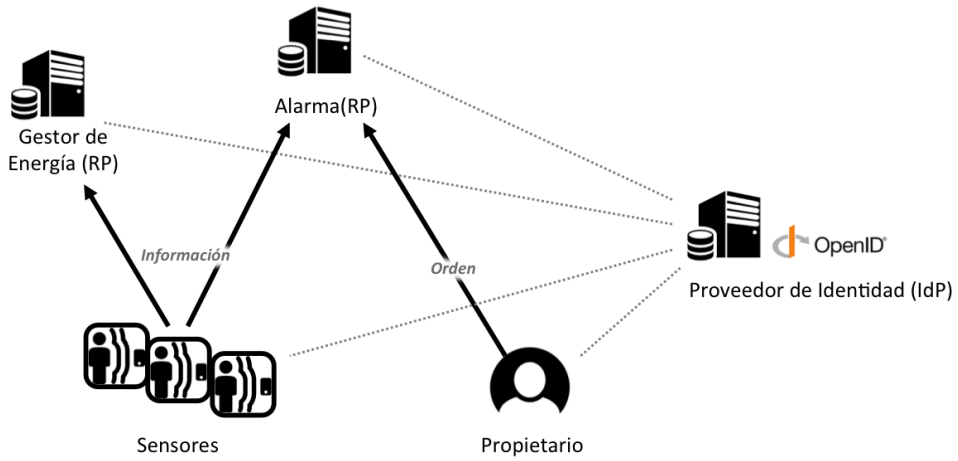

Figura 10: Escenario IoT *Smart Home*

Los sensores no tienen necesidad de identificarse ante el Gestor de Energía, puesto que la información que le proporcionan no tiene importancia especialmente relevante. Sin embargo, si han de identificarse ante la centralita de la alarma en caso de detectar presencia cuando la alarma está activada. Por otro lado, el propietario tiene que identificarse siempre que quiera activar o desactivar la alarma.

5.1.3. Escenario: *Smart City* & *Smart Mobility*

Supóngase ahora una *Smart City* en la que la mayor parte de los vehículos que circulan por ella son vehículos conectados. La ciudad, para canalizar el tráfico por sus principales vías de forma eficiente, dispone de un sistema de gestión de tráfico para informar a cada vehículo de la ruta más eficiente para cada uno. Para ello, el sistema requiere saber el destino, el modelo del vehículo y el la autonomía si fuera eléctrico y, por motivos de seguridad del sistema, la matrícula del vehículo, aunque esta información no se utiliza la utiliza para preservar la privacidad de los usuario. Esta información es verificada por diferentes IdPs pertenecientes a las marcas de vehículos que, si bien no deben mantener una relación de confianza preestablecida

con el sistema de gestión de tráfico, pueden negociar la confianza utilizando el intercambio de atributos de OpenID.

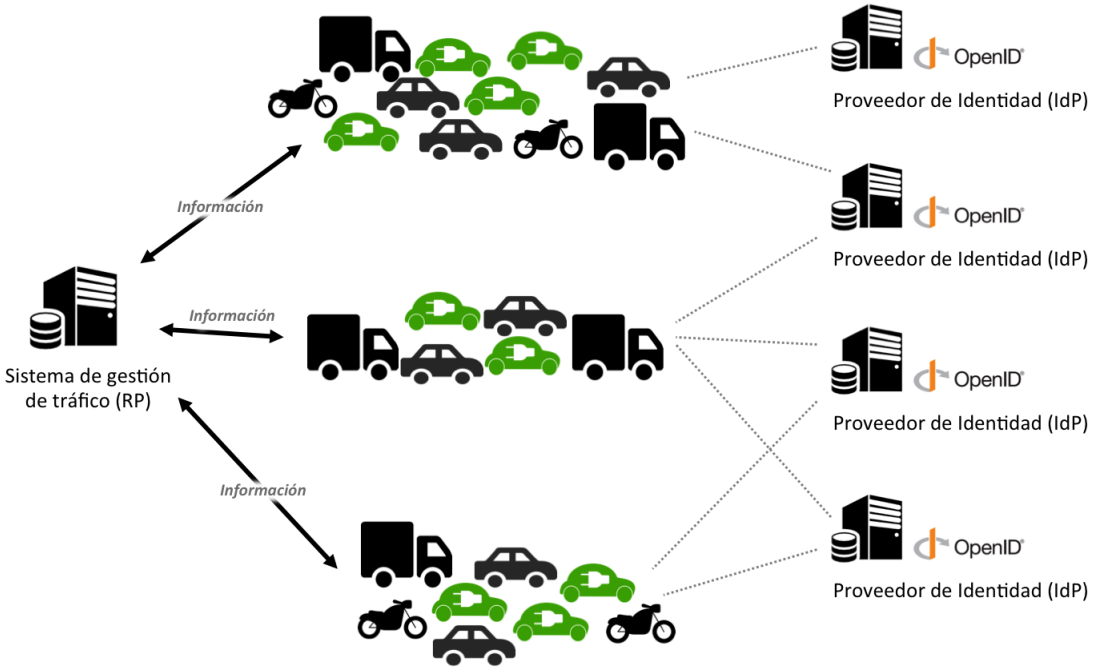

Figura 11: Escenario IoT *Smart City & Smart Mobility*

Cada vehículo informa al sistema de gestión de tráfico con la información que requiere identificándose cada cual con su proveedor de identidad correspondiente, y el sistema le propone a cada uno la ruta más eficiente para llegar a su destino en el menor tiempo y con el menor gasto posible.

5.1.4. Escenario: *Smart Building*

Como último escenario, se plantea un hotel que cumple con el concepto de *Smart Building*. El cliente llega al hotel y recibe en su *smartphone* un identificador para utilizar durante su estancia. Con este identificador puede abrir la puerta de su habitación además de utilizar todos los dispositivos de su interior, como la televisión, el termostato, la iluminación o utilizar el servicio de habitaciones. Además, los clientes tienen acceso a un servicio de información turística que el hotel pone a su disposición.

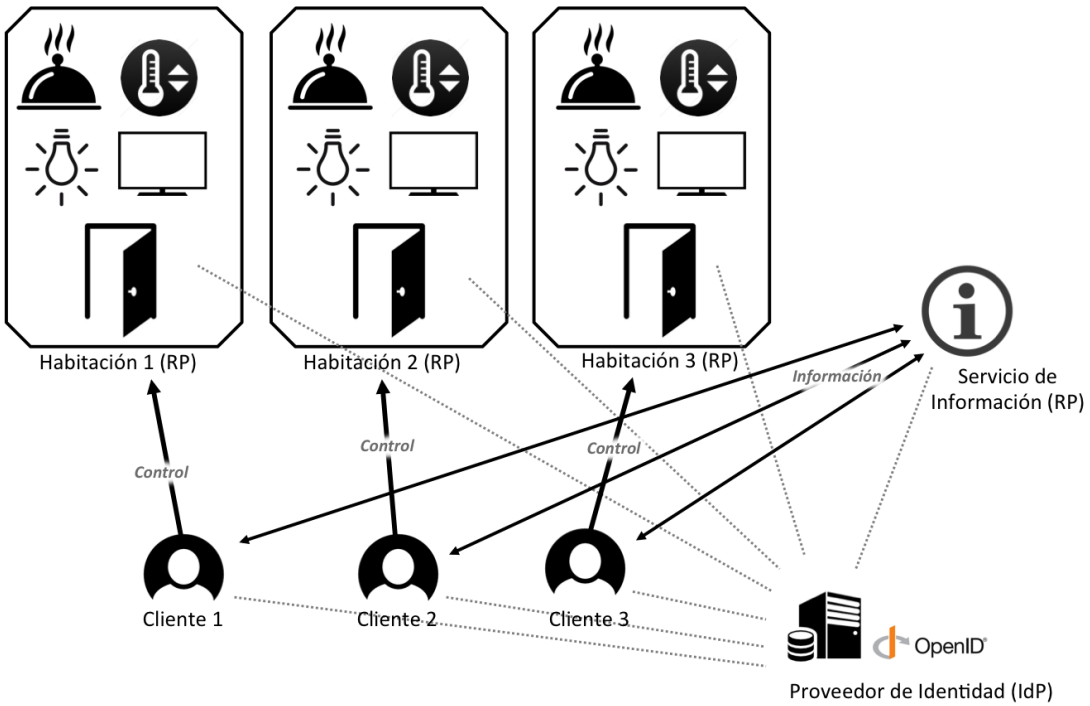

Figura 12: Escenario IoT *Smart Building*

Cada cliente tiene únicamente acceso a abrir la puerta de su habitación y utilizar los objetos de su habitación utilizando el identificador proporcionado y autenticándose en el IdP del hotel, por lo que si intentara abrir otra habitación o usar dispositivos de otras habitaciones le sería denegado el acceso.

5.2. Resultados

Se han llevado a cabo pruebas realizando simulaciones algunos de los entornos propuestos en este proyecto. Las pruebas realizadas son de carácter cualitativo donde se ha podido valorar el correcto funcionamiento de la integración del protocolo OpenID con Akka.

En las pruebas llevadas a cabo se han obtenido los siguientes resultados satisfactorios:

- El proceso de identificación funciona correctamente.
- El intercambio de atributos funciona correctamente.
- Se rechaza el acceso a servicios sin previa autenticación.
- Se rechaza el acceso a servicios de los elementos que no tengan permiso para acceder.
- Se soportan múltiples usuarios, servicios e IdPs

- Se soportan múltiples acciones, cada una con permisos específicos

Debido a limitaciones de tiempo y recursos no han podido realizarse pruebas cuantitativas para medir rasgos como el número de mensajes intercambiados, carga de la red o tiempos de respuesta y realizar pruebas en entornos realmente distribuidos y escalables con un número elevado de actores.

5.3. Análisis objetivo del uso de OpenID en un entorno de IoT real

Se ha demostrado en este proyecto la importancia de que los entornos de IoT incorporen mecanismos de seguridad y, en concreto, mecanismos de identificación para el control de acceso a servicios. Actualmente Internet de las Cosas carece de este tipo de mecanismos, siendo manifiestamente insuficiente el nivel de seguridad de los entornos, por lo que la integración del protocolo OpenID a un entorno real de IoT sería un gran avance en la seguridad del control de acceso a las *cosas*.

Sin embargo, la adopción del protocolo OpenID no sería un paso trivial. Deberían darse una serie de requisitos para su uso en un entorno real cumpliendo con las expectativas planteadas en este proyecto:

1. Los fabricantes deberían ponerse de acuerdo en la adopción de OpenID como estándar de identificación en IoT, de forma que todos los dispositivos tuvieran compatibilidad y libertad de comunicación con otros dispositivos y entornos. Esto supondría para Internet de las Cosas un paso hacia la horizontalidad y estandarización. La adopción de otro protocolo de identificación como estándar distinto de OpenID sería igual de aceptable, si cumpliera con los requisitos establecidos en este proyecto.

2. Los usuarios, además de las cosas, tendrían que disponer de uno o varios identidades emitidos por uno o varios proveedores de identidad, de forma que fueran capaces de aportar distintas identidades en diferentes contextos.

3. Todas las *cosas* tendrían que disponer constantemente de una conexión a Internet para poder verificar las credenciales de quien intente acceder a ellas.

Esto no sería una tarea fácil, y requeriría de varios años para que pudiera verse en un entorno real. Además, antes de que se llevara a cabo, sería necesario realizar un estudio más exhaustivo para analizar la escalabilidad, flexibilidad y robustez de OpenID y terminar de desarrollar el protocolo para su adopción en Internet de las Cosas de forma que soportara las necesidades de cualquier entorno y dispositivo.

6. Desarrollo del proyecto y presupuesto

6.1. Desarrollo del proyecto: diagrama de Gantt

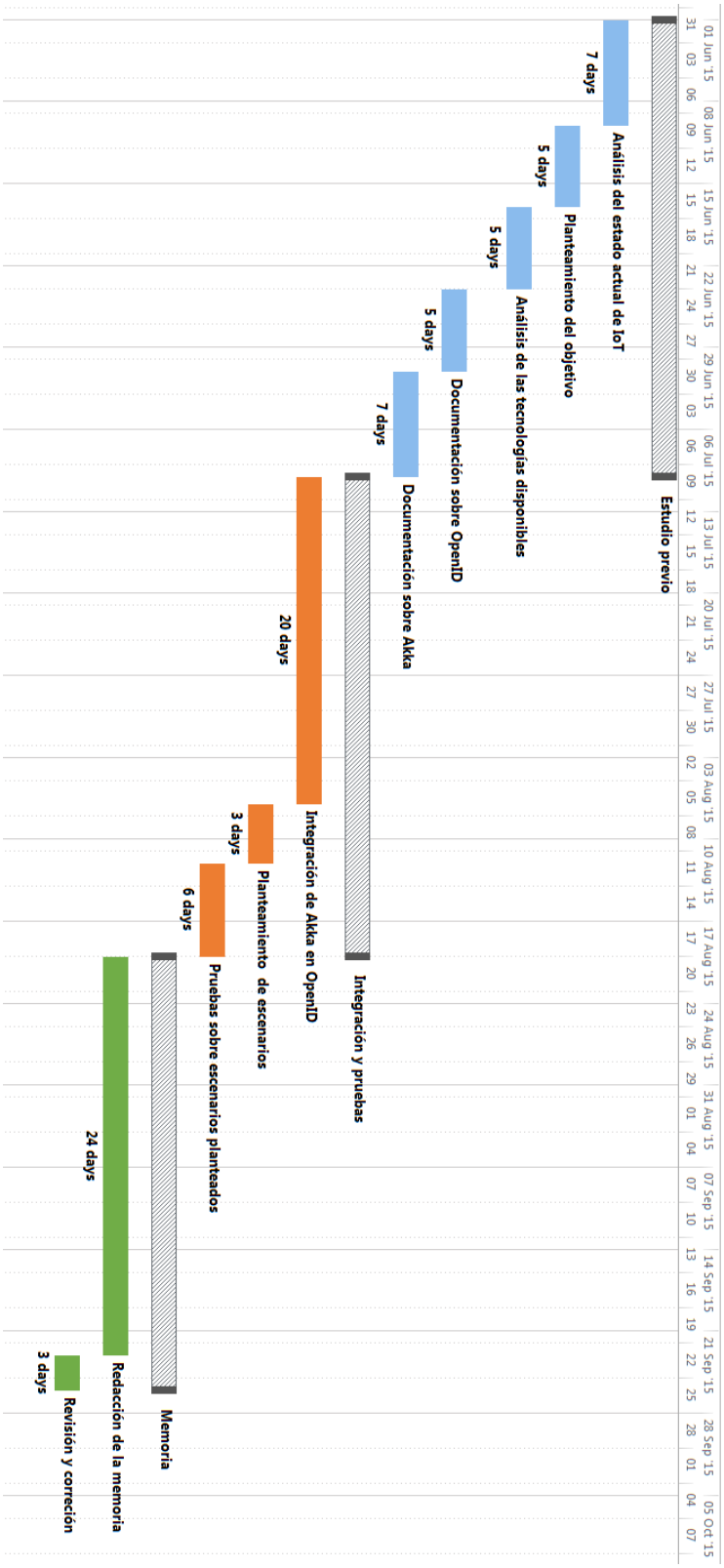

Figura 13: Diagrama de Gantt del proyecto: tareas y tiempos

6.2. Presupuesto

En la siguiente tabla se detallan los costes totales del proyecto:

Materiales	Amortización ordenador			300 €
	Conexión a Internet	3 meses		147 €
RR.HH.	Ingeniero Junior	430 h	25 €/h	10,750 €
	Ingeniero Senior	30 h	90 €/h	2,700 €
TOTAL				**13,897 €**

Tabla 1: Presupuesto total del proyecto

6.3. Entorno socioeconómico y marco legal

Con la expansión de las Tecnologías de Información y la economía digital, los ciudadanos han perdido capacidad para controlar sus datos personales y mantener su privacidad. En enero de 2012 la Comisión Europea comenzó un proceso de reforma de las normas comunitarias para la protección de los datos personales dentro de la Unión Europea [20].

El desarrollo del Internet de las Cosas podría verse frenado y entorpecido en la UE debido a la existente y futura regulación en materia de privacidad de datos. Por tanto, es importante tener en cuenta el marco legal tanto comunitario como específico de cada país a la hora de diseñar y desarrollar nuevos entornos inteligentes, respetando la protección de datos y la privacidad de los ciudadanos en todo momento.

7. Conclusions

7.1. Project conclusions

This project presents a solution to resolve some of the biggest security problems that the Internet of Things currently has. By using the digital identity protocol OpenID, IoT environments are provided with an access control mechanism to *things*, what adds security by means of identification, authentication and authorization.

The development of a software simulation platform, using Akka actor model, to tests simulated environments has allowed for checking the correct functioning of OpenID technology applied to the Internet of Things. The utilization of digital identity technology is a viable option for being applied to IoT environments, since it gives service providers an access control mechanism that is simple, light, efficient and open that can be applied to any device and any environment where services have their own freedom to ask for any kind of attribute that they require to authorize the access.

However, there are some characteristics that have not been checked in this project and they would be a key factor for a possible real implementation of OpenID on real IoT environments, such as scalability and flexibility to work in open environments.

In conclusion, digital identity technology, and in particular OpenID, can provide the Internet of Things with essential security services for access control to devices or *things* of IoT environments that are currently completely vulnerable, solving part of security deficiencies that many devices has nowadays. With the solution given in this project, an attacker that commands a light bulb to switch off, it will first ask for an identification from IdP of the house or another reliable IdP, what resolves the initial problem where this project started.

It is necessary to take measures to protect privacy and security of users and systems that will be very widespread in some years if estimations are fulfilled, and OpenID, an open digital identity technology, is suitable for covering part of that necessities.

7.2. Future lines of work

As future lines of work on access control and digital identity technology for the Internet of Things, some objectives are proposed:

- To develop the concept of *risk* introduced by Identity 3.0 [12] and implement it within a digital identity protocol such as OpenID.
- To analyze and implement the use of long duration environment authentication to reduce network load and IdPs load in environments with high density of users.

- To analyze and implement the mobility of users between different environments and federations.
- To develop a flexible attribute exchange that can understood by all things to make attributes the core of digital identity. That would be linked to the study of a taxonomy that could be applied to any environment.

Referencias

[1] G. Baldini, "Internet of things privacy, security and governance," in *IoT: Converging Technologies for Smart Environments and Integrated Ecosystems* (O. Vermesan and P. Friess, eds.), pp. 222–239, River Publishers, 2013.

[2] Ericsson, "More than 50 billion connected devices," *Ericsson White Paper*, vol. 284, 2011.

[3] S. Goedertier, S. Ackx, K. Dervojeda, J. Devloo, S. Goedertier, L.-D. Hostyn, E. Rouwmaat, G. Vanhaver, O. Verack, and A. Ziemyte, "Benchmark study for large scale pilots in the area of the internet of things," final study report, European Comission, Mar 2015.

[4] H. Security, "Internet of things research study," tech. rep., HP, Jul 2014.

[5] "Así pueden "hackearte" cualquier aparato conectado a internet," Jul 2015. `http://tecnologia.elpais.com/tecnologia/2015/07/10/actualidad/1436539664_188672.html`.

[6] G. Gang, L. Zeyong, and J. Jun, "Internet of things security analysis," in *Internet Technology and Applications (iTAP), 2011 International Conference on*, pp. 1–4, Aug 2011.

[7] M. Muñoz Organero, *Introducción a la seguridad*, ch. 2. UC3M, 2015.

[8] O. Mazhelis, H. Warma, S. Leminen, P. Ahokangas, P. Pussinen, M. Rajahonka, R. Siuruainen, H. Okkonen, A. Shveykovskiy, and J. Myllykoski, "Internet-of-things market, value networks, and business models: State of the art report," tech. rep., University of Jyväskylä, 2013.

[9] K. Ashton, "That 'Internet of Things' Thing," Jun 2009. `http://www.rfidjournal.com/articles/view?4986`.

[10] Expert Group on the Internet of Things, Sub-Group on Identification, *Conclusions of the Internet of Things public consultation*, ch. 6. Internet of Things Fact Sheet Identification. European Commission, 2012.

[11] D. Díaz-Sánchez, *Introduction to Access Control Systems*, ch. 2. UC3M, 2011.

[12] Global Identity Foundation, "Identity 3.0 principles," Jun 2014. `http://www.globalidentityfoundation.org/downloads/Identity_3.0_Principles_v1.1.pdf`.

[13] OpenID Foundation, "Openid authentication 2.0 (specification)," Dec 2007. `http://openid.net/specs/openid-authentication-2_0.html`.

[14] G. A. Agha, "Actors: A model of concurrent computation in distributed systems," tech. rep., Massachusetts Institute of Technology, 1985. `http://hdl.handle.net/1721.1/6952`.

[15] J. Bonér, D. Farley, R. Kuhn, and M. Thompson, "The Reactive Manifesto," Sep 2014.

[16] V. Akman, "Book Review: "Actors: A Model of Concurrent Computation in Distributed Systems"," 1990.

[17] Typesafe Inc, "Akka java documentation, release 2.3.9," Jan 2015.

[18] "openid4java." `https://github.com/jbufu/openid4java`.

[19] "openid4java: Sampleserver.java." `https://github.com/jbufu/openid4java/blob/master/src/org/openid4java/server/SampleServer.java`.

[20] "Protection of personal data." `http://ec.europa.eu/justice/data-protection/`.

Página en blanco.

Apéndice A Extended summary

A.1 Introduction

The way we currently face day-to-day and the way we interact with daily object will experience a huge change with the arrival of *Smart Environments* plenty of sensors and hyperconnected devices. The so-called *Internet of Things* is about to arrive, and it brings improvement of quality and efficiency of services thanks to other complementary technologies as *Big Data* and *Cloud Computing* that will set Internet of Things as the core of economy and industries.

Expert estimations expect that more than 50 billion devices will be connected by 2020, what implies many challenges to settle the basis for the Internet of Things. Current network communications will have to adapt to such growth, and IoT will require systems to be highly distributed and decentralized and extremely flexible.

On the other hand, there are other important issues regarding privacy and security that must be treated as soon as possible. Some studies reveal that current connected devices present an alarming number of vulnerabilities such as gathering unnecessary personal information, transmit information through a network without encryption, weak passwords and insufficient authentication and authorization mechanisms. This vulnerabilities are an actual risk for users and systems security.

The motivation of this project is to meet the need of security that the Internet of Things has right now to establish a solid and secure base to settle the technology and protocols that will dominate the IoT environments.

In broad strokes, the main objective is to improve security services and communication protocols for the Internet of Things and provide devices with such security services, to build Smart hyperconnected reliable environments. Basic security requirements are: confidentiality, authenticity, integrity, availability, access control and non-repudiation. Additionally, due to the scale of the Internet of Things, systems must also fulfill with scalability, flexibility, robustness and lightness.

More concretely, this project will focus on access control to provide the Internet of Things environments with mechanisms that allow for establishing a policy on access control by *digital identity* technology. As implementing that in a real IoT environment is beyond the possibilities of this project, a software simulation platform will be developed to test digital identity functioning on simulated environments.

A.2 State of the Art

Identity systems

In the Internet of Things, a *thing* is a physical or virtual intelligent entity

with capacity to communicate with other things to achieve one or several aims. The Internet of Things is a concept that integrates digital connected things into real physical world. It represents the convergence of several technologies: *RFID* (Radio-Frequency Identification), a technology for automatic object identification by means of RFID tags; *M2M* (machine-to-machine) communication, that allows autonomous communication between autonomous systems; *Ubiquitous Computing*, a concept in which computing surrounds personal environments; and *WoT* (Web of Things), which consists on the utilization of current Web standards to access electronic devices.

This technologies have been used during last decade individually in different industries like automotive, home automation and health services.

The Internet of Things suppose a challenge by itself, since smart objects will take decisions that may have negative consequences if they fail. Having environments with many connected devices implies that they can be object of cyberattacks, what means that environment security and users'privacy may get compromised. For that reason, information must be protected against non-authorized access and must be encrypted when transmitted.

An *identity* can defined as the set of attributes that an entity says about itself and others says about it. Identifying things in IoT must accomplish with some requisites: discovery mechanism, mobility of devices, dynamism, scalability, simplicity, efficiency, interoperability, flexibility, extensibility, reliability and security.

Access control involves different security processes: *authentication*, what implies identification, is the process in which it is verified that an entity is what it says to be; *authorization*, to determine if an entity has permissions required to access; *policy enforcement*, that checks the access policy to accessed resource; and *accountability*, to log access control process.

There are different systems that can provide access control to a given services. Traditionally, the most common access control system in Web services is centralized on the services, which is isolated and has the control over the whole process of

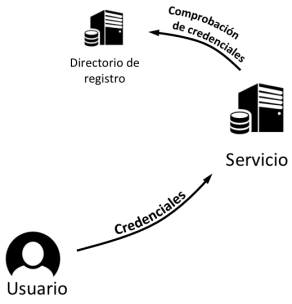

Figura 14: Centralized access control

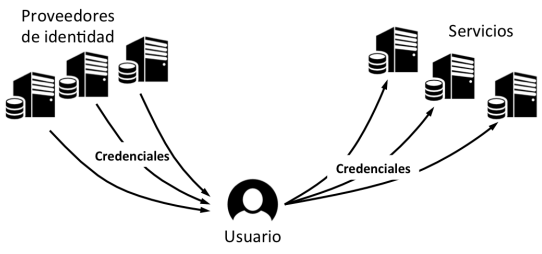

Figura 15: Distributed access control

identification usually by means of user-password pair (Figure 14). In a distributed access control, several identity providers can supply a user with credentials to use them in any services as the user wants (Figure 15). Identity 2.0 supports different ways of authentication between user an providers to give user access to any site that relies on its identity provider giving scalability and flexibility to the system.

OpenID is an open distributed authentication protocol that allows users to authenticate in any site that supports the standard by means of an URI identifier. One user may have several identifiers that can be used in any site that allows OpenID authentication. This identifiers can provide any kind of information by using attributes.

Figura 16: OpenID logo

In OpenID there are three entities that participate in the authentication process: *end-user*, the one that wants to access a service; *Relying Party* (RP), the service the user wants to access; and *OpenID Provider* or *Identity Provider* (IdP), the server that provides the user with an identifier.

The process of authentication has the following steps:

1. User access a service and supplies an identifier to RP.
2. RP discovers IdP.
3. RP and IdP establish a secure association.
4. RP redirects the user to IdP.
5. User authenticates to IdP.
6. IdP redirects the user to RP.
7. RP verifies the authentication.

Actor model

The *actor model* is a model of computation based on entities called actors that are concurrent, independent, distributed and dynamic. Actors have capability to communicate with other actors using asynchronous messages that they receive in their associated mailbox. When an actor processes a message, it can react by changing its local state, messaging other actors or creating new actors.

These properties make actor model ideal for distributed systems, as actors are independent and can communicate regardless their physical location and they can

adapt to work load by creating new actors. Thus, actor model provides concurrency, distribution, flexibility and scalability.

Akka is an open source framework for Scala and Java that implements actor model. It provides an abstraction layer for programming reactive applications, transparency regarding actor distribution and high fault tolerance thanks to *let-it-crash* philosophy.

Figura 17: Akka logo

In Akka, each actor has its own reference, state, behavior, mailbox, actor children and strategy supervisor and they are organized in hierarchy. When an actor receives a task, it can divide it into smaller tasks to delegate them on its children.

A.3 Design of a simulation environment for the Internet of Things

Internet of Things will have a great variety of devices and functions that will create a very heterogeneous network as much in functionality as in computation capacity. The requisites for the development of a simulation environment for the Internet of Things are the following:

- Concurrency
- Capacity of connection and communication for any device
- Autonomous devices
- Auto-discovery mechanism
- Simplicity in digital identity
- Exportability to more complex identity systems
- Non-defined network topology
- Decentralized and flexible

To model Internet of Things in a simulation, it must accomplish with concurrency, since many things will be functioning and communication at the same time; heterogeneity, as devices may have very different properties; and undefined topology, since it has to be flexible and scalable and it must adapt to new devices. This characteristics are all collected in actor model, what is a good tool for the purpose of this project.

Moreover, it is expected that IoT environments are responsive and can respond quickly and effectively to any problem on the system. Also, things might move between different environments, what implies that it must support mobility of actors. Finally, one thing may have more than one application or functionality, what can be easily modeled with one actor for each application.

Among all actor model implementations, the reasons why Akka has been chosen for this project are:

1. Open source
2. API for Java
3. Complete and detailed documentation
4. Easy use
5. Big community
6. Well implemented model
7. Very used in real projects

On the other hand, it is necessary to guarantee security on access control to devices, resources and applications. To distinguish between authorized and non-authorized accesses it can be used digital identity technology. Digital identity must also fulfill some requirements such as being flexible and open to free exchange of attributes and being light and simple enough such that devices with less computing capacity can use it.

The digital technology chosen for this project is OpenID for the following reasons:

1. Open standard
2. Well-known protocol in Web
3. Simple and sufficient
4. Light solution

In particular, it has been used *openid4java* libraries, that is an open implementation of OpenID protocol for Java.

The architecture that the simulation platform will have the following fundamental elements:

- *User*: human entity that can interact with other network elements and make use of services that they provide.
- *Actuator*: device with capacity to take autonomous decisions and can control other devices by sending messages.
- *Sensor*: device that measures some environment variables to serve it as information to users, actuators or services.
- *Relying Party* (RP): service that users and actuators access in a secure way after authenticating their identity.
- *Identity Provider* (IdP): network element responsible for authenticating users and devices that want to access a service that requires identification.

A.4 Development of a simulation environment

To model a thing with an actor it must have a set of characteristics that integrates it in an environment and shapes its functionality. A thing has the following attributes:

- *Identity* (URI): a public reference used by other actors to interact with it. It has the following form: `akka://actor-iot-system/thingname`
- *IdP address*: a thing knows the address path of its IdP. Though it may have several IdP for several identities, the simulator has been implemented with one identity per thing.
- *Entity properties*: set of attributes that describe the thing and its properties. It is expected to be flexible and open, but it has been implemented with a short fixed set of properties for testing purposes.
- *Capabilities*: set of attributes that describe what can this thing do. As in the previous field, it has been implemented for testing purposes with a short fixed set.

To integrate OpenID protocol within an IoT environment, OpenID elements has been adapted to IoT things. Thus, users in OpenID are mapped to users and different things that use other thing services (RP). Using *openid4java* libraries, they have been implemented to follow OpenID protocol flow.

Since OpenID was designed for Web, it works using HTTP and HTTPs transport protocols, and so is implemented in *openid4java* libraries, that has been modified step by step to integrate Akka transport protocol such that it can be used by actors in the simulation platform to send serialized messages to other actors.

Thus, every module of *openid4java* libraries that used HTTP protocol for sending any message or that operated with the class `UrlIdentifier`, that did not allow `akka` as transport protocol, has been modified to integrate it in OpenID protocol. Some of the modified classes are `UrlIdentifier`, `Discovery` and `ConsumerManager`, which contain most of the process that used HTTP as protocol.

A.5 Scenarios and tests

Internet of Thing is a general concept that covers all fields. This project consider some scenarios on which digital identity would be relevant:

1. *Generic scenario*, that is a generalization of any real scenario, since any scenario is composed by four type elements: user/actuator, sensors, services, identity provider.
2. *Smart Home*, in which there are a security alarm system that can be activated by the owner and triggered by motion sensors, that also cooperate with an energy management system.

3. *Smart City & Smart Mobility*, that is an scenario of a city in which connected vehicles are suggested by a traffic management system the best route to get their destinations in the shortest time with the maximum efficiency.

4. *Smart Building*, consisting on an hotel where all room objects are connected devices, even room doors that can be opened using an identity that each guest receives in his smartphone. Guests have available a tourist information service that can also be accessed with identity provided by the hotel.

Qualitative tests have been carried out using the simulation platform and results show the correct functioning of OpenID over Akka simulated environments. Identification process works correctly and attributes are successfully exchanged. Accesses without authentication are rejected, as well as authenticated accesses that do not have permission to use a service. Simulation environment also support multiple users, services, IdPs and multiple actions, each of that has specific permissions.

Due to time and resources limitation, it has not been possible to carry through quantitative tests to measure variables like number of exchanged messages, network load, times of response and to perform actual distributed and scaled simulations.

It has been proved that Internet of Things has a huge need in security and that OpenID protocol works in simulated environments and it can provide IoT with access control security service through identification and authentication. However, for OpenID to be implemented in real environments, some previous work would be necessary. Firstly, it would be necessary to perform a more exhaustive study to analyze the actual scalability, flexibility and robustness of OpenID in the real Internet of Things. Additionally, manufacturers would have to agree the standardization of OpenID as digital identity protocol for IoT, such that any thing can talk to other things and identify itself. Also, users would have to get digital identities from several IdPs to use them in different environments.

A.6 Conclusions

This project presents a solution to resolve some of the biggest security problems that the Internet of Things currently has. By using the digital identity protocol OpenID, IoT environments are provided with an access control mechanism to *things*, what adds security by means of identification, authentication and authorization.

The development of a software simulation platform, using Akka actor model, to tests simulated environments has allowed for checking the correct functioning of OpenID technology applied to the Internet of Things. The utilization of digital identity technology is a viable option for being applied to IoT environments, since it gives service providers an access control mechanism that is simple, light, efficient and open that can be applied to any device and any environment where services have their own freedom to ask for any kind of attribute that they require

to authorize the access.

However, there are some characteristics that have not been checked in this project and they would be a key factor for a possible real implementation of OpenID on real IoT environments, such as scalability and flexibility to work in open environments.

In conclusion, digital identity technology, and in particular OpenID, can provide the Internet of Things with essential security services for access control to devices or *things* of IoT environments that are currently completely vulnerable, solving part of security deficiencies that many devices has nowadays. With the solution given in this project, an attacker that commands a light bulb to switch off, it will first ask for an identification from IdP of the house or another reliable IdP, what resolves the initial problem where this project started.

It is necessary to take measures to protect privacy and security of users and systems that will be very widespread in some years if estimations are fulfilled, and OpenID, an open digital identity technology, is suitable for covering part of that necessities.

Apéndice B Introducción

En los últimos años, ha ido sonando con mayor fuerza el concepto de los *Smart Environments*: Smart Cities, Smart Grid, Smart Home, Smart Mobility, Smart Building... Aunque actualmente son conceptos en fases de pruebas, será cuestión de poco tiempo el que estos conceptos comiencen a verse de forma extendida.

Nuestra forma actual de afrontar el día a día y de interactuar con objetos cotidianos cambiará sustancialmente con la integración de dispositivos electrónicos con el mundo físico. La sensorización del entorno y la hiperconectividad entre dispositivos dan forma al llamado *Internet de las Cosas* o *IoT*, del inglés *Internet of Things*, en el que las cosas son conscientes de la situación que las rodea y toman decisiones inteligentes persiguiendo un fin determinado [1].

El ecosistema de Internet de las Cosas está emergiendo rápidamente. Según las estimaciones de expertos, se espera que para el año 2020 haya más de 50 billones[1] de dispositivos conectados siguiendo una curva de crecimiento exponencial [2]. El número de conexiones entre dispositivos y, especialmente la comunicación *machine-to-machine* (M2M) crecerá de igual manera. Esto supone un reto para las redes de comunicaciones actuales y ha de cuestionarse la topología de las redes del futuro. Sistemas distribuidos, altamente descentralizados y con gran flexibilidad para adaptarse a cambios venideros serán requisitos necesarios para el despliegue de IoT.

La inmensa cantidad de datos generados por estos sensores y dispositivos será objetivo del *Big Data* y *Cloud Computing* para extraer información útil para mejorar la calidad y eficiencia de los servicios a través del análisis de datos. IoT será el eje sobre el que se centrarán las demás industrias en busca de reducir costes, optimizar la utilización de recursos, mejorar su eficiencia y aumentar su productividad añadiendo valor de forma transversal al resto de sectores.

Sin embargo, aunque nadie duda del increíble potencial de IoT en un futuro, aún debe madurar y superar numerosos retos que asienten su crecimiento. El despliegue actual de entornos IoT se caracterizan por seguir una estructura vertical y utilizar soluciones propietarias que limitan la interoperabilidad con el resto del ecosistemas [3]. La falta de consenso sobre estándares y protocolos y la incertidumbre sobre la dirección que tomará la evolución de IoT no facilita la expansión.

Por otro lado, se plantean cuestiones respecto a la privacidad y a la seguridad. En un entorno repleto de sensores y dispositivos recogiendo y procesando información de cualquier índole de forma constante es de suma importancia poder asegurar que estos datos se mantienen fuera del alcance terceros no autorizados, ya sean personas o cosas, y que se utilizan de forma ética para fines previamente consentidos, garantizando la privacidad de los usuarios.

Este proyecto se centra en el problema de la seguridad del Internet de las Cosas, más concretamente en la identificación de las cosas. Los dispositivos han

[1] 50.000 millones

de ser capaces de distinguir entre comunicaciones legítimas e ilegítimas, y poder identificar e identificarse ante otros para interactuar de forma segura con el entorno y garantizar un intercambio seguro de información.

B.1 Descripción del problema

El Internet de las Cosas ya ha llegado, aunque aún esté en una fase prematura, y una de las mayores debilidades de los dispositivos que comienzan a conformarlo es la seguridad. Un estudio llevado a cabo por HP en julio de 2014 [4] analizó 10 dispositivos de los más comunes en IoT como TVs, termostatos, enchufes inteligentes, sistemas de riego, cerraduras inteligentes o hubs de control de múltiples dispositivos, y los resultados revelaron un alarmante número de vulnerabilidades.

El 80 % de los dispositivos recababan información personal, y muchos de ellos transmitían esta información por la red local e internet sin ningún tipo de cifrado, lo que supone un riesgo para la privacidad cuando la información transmitida es sensible y confidencial. Además, también el 80 % de los dispositivos analizados presentaban vulnerabilidades por contraseñas débiles y mecanismos de autenticación y autorización insuficientes.

Está debilidad queda patente en este artículo [5], en el que la empresa Sophos demuestra cuán fácil es acceder al control de dispositivos conectados. Un atacante que consiga introducirse una red WiFi, puede tener control absoluto sobre las *cosas* conectadas a la misma red local. Con tan solo ejecutar un script, puede encender y apagar una bombilla a su antojo sin ningún tipo de autenticación ni control de identidad. Este tipo de vulnerabilidades puede presentar un riesgo cuando los dispositivos conectados a la red son más relevantes o cuando controlan aspectos importantes del entorno. Por ejemplo, en una casa podría suponer un riesgo que el acceso al control de la caldera o de la alarma de seguridad fuera tan fácil como se muestra en el artículo citado. En otros entornos, como en una *Smart City* sería un peligro si el control de los semáforos de un cruce se viera comprometido de tal manera, o si dispositivos de *Smart Grid* pudieran alterarse con la misma facilidad.

B.2 Motivación

Este proyecto surge debido a la deficiencia patente en servicios de seguridad en el Internet de las Cosas en todos sus aspectos: confidencialidad, integridad, autenticidad, control de acceso.

Como se ha demostrado, hay mucho trabajo pendiente en tema de seguridad en el Internet de las Cosas. Por suerte, aún hay margen de maniobra, y hasta el año 2020 no tendremos los 50 billones de dispositivos conectados, recogiendo y compartiendo información por la red y controlando múltiples aspectos de nuestro entorno y nuestras vidas. Es importante establecer unos cimientos sólidos y *seguros* sobre los que asentar la tecnología y los protocolos de comunicación y control que veremos a lo largo de los próximos años.

La motivación de este proyecto es cubrir las necesidades de seguridad que tiene actualmente Internet de las Cosas debido a que es una gran barrera para su desarrollo y expansión, además de suponer un peligro para la privacidad de los usuarios y la seguridad de la sociedad. Es importante que los todos los dispositivos y objetos conectados en el futuro puedan ofrecer garantías en términos de seguridad digital.

B.3 Objetivos

En una sociedad en la que las tecnologías de la información tienen cada vez más importancia, se pueden establecer dos tipos de objetivos a grandes rasgos. Por un lado, un objetivo social de concienciación y educación de la sociedad en las Tecnologías de la Información y, por otro lado, un objetivo técnico para la mejora de todos los aspectos de la seguridad en servicios y protocolos de comunicación de Internet de las Cosas. Ambos objetivos son necesarios y se complementan.

La educación de la sociedad en las Tecnologías de la Información es algo que se ha empezado a inculcar en las nuevas generaciones con la aparición de las nuevas tecnologías. Sin embargo, la consciencia del la seguridad no se ve reflejada en la adopción de las nuevas tecnologías en el día a día. Existe mucha desinformación en la sociedad debida a la falta de formación que provoca que el usuario medio esté en una situación de inseguridad en la red. Acciones habituales como la elección de una contraseña o navegar por la Web pueden llevar al usuario a comprometer su privacidad si no se hace de forma adecuada. En este sentido, es necesario un compromiso de las autoridades para fortalecer la educación en la seguridad de un pilar básico de nuestra sociedad como son las Tecnologías de la Información.

Por otro lado, dotar a los dispositivos de todos los servicios de seguridad es importante para crear entornos *Smart* hiperconectados e inteligentes seguros y fiables. Toda red ha de cumplir con unos requisitos de seguridad para almacenar, transmitir y procesar información [6] [7]:

- Confidencialidad: prevenir que la información sea accesible por terceros no autorizados.
- Autenticidad: garantizar que el origen de la información está correctamente identificado.
- Integridad: garantizar que la información transmitida no ha sido modificada por terceros no autorizados.
- Disponibilidad: asegurar que la información esté disponible cuando entidades autorizadas lo requieran.
- Control de acceso: controlar el acceso a la información y a los sistemas.
- No repudio: ni el emisor ni el receptor pueden negar la transmisión.

Además, debido al dimensionamiento que tendrá el Internet de las Cosas, cualquier sistema o protocolo de seguridad debe cumplir otros requisitos adicionales:

- Escalable: garantizar el correcto funcionamiento ante el crecimiento de la red.
- Robusto: garantizar el correcto funcionamiento ante errores en la red, en la transmisión de mensajes o en la información.
- Flexible: capacidad para adaptarse a dispositivos con diferente funcionalidad, capacidad y características.
- Ligero: cualquier dispositivo debe ser capaz de implementar las funcionalidades básicas de seguridad sin verse limitado por su capacidad de cómputo.

Como se ha explicado, Internet de las Cosas integrará los dispositivos electrónicos en el mundo físico, permitiendo a las personas interactuar con lo que nos rodea de forma digital y a distancia, y a las *cosas* interactuar también entre ellas. Sin embargo, es posible que el acceso a ciertos dispositivos deba estar restringido a ciertos usuarios o modelado para permitir diferentes accesos para diferentes roles. Por ejemplo, el termostato de casa puede ser modificado por los propietarios Bob y Alice, pero su hijo Mike solo puede modificar la temperatura de su habitación 2 °C arriba o abajo; cuando tienen invitados, solo tienen acceso al termostato para ver la temperatura de la casa, pero no pueden modificar nada sin permiso de los propietarios; por otro lado, la compañía energética tiene acceso al termostato para recopilar los hábitos de consumo agregados de la casa y, en caso de avería, un técnico puede acceder a la configuración del termostato. Por tanto, un entorno de IoT debe ser multiusuario y multirol.

B.3.1 Objetivos del proyecto

Debido a las limitaciones de tiempo y recursos de los que se disponen para este proyecto, cumplir todos los objetivos expuestos previamente sería difícil. Por ello, de todos los sistemas de seguridad este proyecto se centrará en el *control de acceso*.

El objetivo de este proyecto será dotar a los entornos de Internet de las Cosas de mecanismos que permitan establecer una política de control de acceso a los dispositivos y la información. Para ello, se utilizará la tecnología de *identidad digital* para resolver las deficiencias de seguridad en el control de acceso de forma que, cuando un atacante ordene a una bombilla que se apague (sección B.1, [5]), ésta primero le pregunte quién es él y qué autorización tiene para emitir tal orden.

Como implementar un sistema de identidad digital en un entorno real de IoT queda fuera del alcance de las posibilidades de este proyecto por causas, de nuevo, de tiempo y recursos, se desarrollará una plataforma de simulación por software. El objetivo de este simulador será poder probar el funcionamiento de la identidad digital en entornos simulados. Además, sobre este simulador se podrían probar en el futuro nuevas funcionalidades o realizar pruebas más exhaustivas en diferente entornos simulados para comprobar su rendimiento, carga, coste u otros parámetros de interés.

Apéndice C Conclusiones

C.1 Conclusiones del proyecto

Este proyecto presenta una solución para resolver algunos de los grandes problemas de seguridad que presenta actualmente Internet de las Cosas. Mediante el uso del protocolo de identidad digital OpenID se dota a los entornos de IoT de un mecanismo de control de acceso a las cosas añadiendo seguridad mediante la identificación, autenticación y autorización.

El desarrollo de una plataforma de simulación por software, utilizando el modelo de actores de Akka, donde realizar pruebas con entornos simulados ha permitido comprobar el correcto funcionamiento de la tecnología OpenID aplicada a Internet de las Cosas. La utilización de tecnología de identidad digital ha resultado viable para ser aplicada a entornos de IoT, ya que aporta a los dispositivos que ofrecen servicios un control de acceso sencillo, ligero, eficiente y abierto que puede ser aplicado a cualquier dispositivo y en cualquier entorno, teniendo los propios servicios libertad para solicitar cualquier tipo de atributo que requieran para autorizar el acceso.

Sin embargo, hay ciertas características que no se han comprobado en este proyecto y que serían un factor clave para la posible implementación real de OpenID a entornos IoT reales, como la escalabilidad y la flexibilidad para funcionar en entornos no confinados.

En conclusión, la tecnología de identidad digital y, en concreto OpenID, puede dotar a Internet de las Cosas de servicios de seguridad esenciales para el control de acceso a dispositivos o *cosas* de entornos IoT que actualmente son absolutamente vulnerables, resolviendo parte de las deficiencias de seguridad que presentan una gran cantidad de dispositivos a día de hoy. Con la solución presentada en este proyecto, un atacante que ordene a una bombilla que se apague primero le pedirá que se identifique en el proveedor de identidad de la casa o en algún otro de confianza, lo que resuelve el problema inicial del que se partió para este proyecto.

Es necesario tomar medidas para proteger la privacidad y seguridad de los usuarios y de los sistemas que en unos pocos años veremos de forma extendida en nuestro entorno si se cumplen las estimaciones y OpenID, una tecnología abierta de identidad digital, puede cubrir parte de las necesidades actuales.

C.2 Líneas de trabajo futuro

Como líneas de trabajo futuro en el control de acceso y tecnología de identidad digital para Internet de las Cosas, se proponen varios objetivos:

- Terminar de desarrollar el concepto de *riesgo* introducido en Identidad 3.0 [12] e implementarlo en un protocolo de identidad digital como OpenID.

- Analizar e implementar el uso de autenticación en entornos con duración prolongada para disminuir la carga de red y la carga de los IdPs en entornos de alta densidad de usuarios.
- Analizar e implementar la movilidad de usuarios entre diferentes entornos y diferentes federaciones.
- Desarrollar un intercambio de atributos flexible que pueda ser entendido por todas las *cosas* para hacer de los atributos el núcleo de la identidad digital. Esto iría ligado al estudio de una taxonomía que fuera aplicable para cualquier entorno.