UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA DE TELECOMUNICACIÓN

PROYECTO FIN DE CARRERA

Provisión de servicio de comercio electrónico con soporte a pasarela de pago para verificación del pago a través de terminal lMS

Autor: Manuel Palomino García

Tutor: Daniel Díaz Sánchez

Agradecimientos

A mis padres, que siempre han confiado en mí y me han apoyado en los momentos más difíciles a lo largo de la carrera. Sin vuestra ayuda no habría llegado a ninguna parte.

A Inma, que ha hecho que sea la persona más feliz del mundo desde que estamos juntos y me ha ayudado a levantarme en los peores momentos. Sin ti no podría seguir adelante.

A mis compañeros de clase: Bea, Carlos, Dani, Fer, Javi, Javo, Kiko, Luis, Miki, Natalia, Raúl, Rosa, Rober y Santi, por compartir conmigo todos estos años y ayudarme siempre que lo he necesitado. También a todos aquellos que he ido conociendo a lo largo de la carrera y que han formado parte de esta etapa de mi vida.

A todos mis amigos de El Escorial, por mostrarme su apoyo en todo momento y mantener la amistad que nos une desde tantos años.

A mis compañeras de piso, Lydia, Ortiz y María, por haberme hecho sentir tan a gusto estos últimos meses y animarme cuando me sentía mal.

A mi tutor, Dani, por su ayuda, consejos y aclaraciones durante todos estos meses, y estar siempre disponible para echarme una mano cuando he tenido problemas. Muchas gracias por tratarme tan bien y ser paciente conmigo.

A Davide, por ayudarme con los problemas que me han surgido a lo largo del proyecto y haber estado siempre disponible para ello.

Resumen

En los últimos años el uso de Internet y la realización de transacciones económicas a través de dicha red han crecido considerablemente. La utilización del comercio electrónico permite al usuario realizar las transacciones desde cualquier lugar, sin necesidad de desplazarse a la tienda en la que desea realizar la compra y ahorrar el dinero y el tiempo que requeriría dicho desplazamiento.

Sin embargo, la utilización del comercio electrónico para la realización de compras por Internet no goza de un respaldo mayoritario debido a la desconfianza de los usuarios a proporcionar sus datos bancarios o la información de sus tarjetas de crédito a través de la red.

En este proyecto, se han estudiado los diferentes métodos de pago electrónico existentes en la actualidad, y su posible mejora para la provisión de un servicio de pago electrónico fiable y seguro para los clientes. La futura convergencia entre redes de comunicación ha originado el planteamiento de un sistema de pago que permita al usuario realizar la compra a través de la Web pero validarla a través de un terminal IMS (IP Multimedia Subsystem).

Más concretamente, este proyecto ha tenido como objetivo el diseño de una aplicación convergente que permita interconectar la Web con las redes de próxima generación para proporcionar servicios de pago electrónico seguros con soporte para la verificación de las transacciones económicas a través de un terminal IMS. Para ello se hace uso de dos canales de comunicación diferentes (Internet y telefonía móvil), lo que dota de una gran robustez al sistema, ya que es complicado comprometer las dos redes simultáneamente.

La necesidad de proteger la información privada de los usuarios hace que el sistema diseñado soporte gestión de identidad. Como consecuencia de ello, los usuarios del sistema pueden controlar la información de su perfil que deseen proporcionar a un determinado proveedor de servicio.

Por otro lado, se ha tratado el problema de la gestión de credenciales por parte de los usuarios. Para ello, se ha proporcionado al sistema desarrollado de un mecanismo de identificación SSO (Single Sign-On), que evita al usuario tener que identificarse repetidamente en los diversos sitios Web a los que acceda.

La solución planteada podría tener una gran proyección, ya que la telefonía móvil goza de una gran aceptación y su uso se encuentra mundialmente extendido. Debido a ello, es lógico pensar que los usuarios podrían encontrarse más confiados a enviar su información privada a través de su teléfono móvil que a través de Internet.

Abstract

Financial transactions have increased in number in the last years. The use of electronic commerce allows users to make transactions from anywhere, without having to go to the store where they want to make the purchase thus, saving money and time.

The lack of a comprehensive security infrastructure has delight in tormenting e-commerce. The use of electronic transactions in a daily basis is far from being a reality. Due to several security problems, customers are reluctant to hand out their private information, especially their financial information as credit cards.

In this thesis, we have studied the different electronic payment methods currently available and feasible improvements towards a safer and reliable service for customers. The future convergence of communication networks has led to the approach of an authorized payment system by mobile phone that allows the user to purchase online but validate it through his IMS terminal.

The system developed will use two different communication channels (Internet and mobile), making hard to compromise both networks simultaneously. The purpose of such development is to find a solution for making safe purchases over the Internet increasing so user confidence in electronic commerce.

On the other hand, the need to protect private information from users, makes necessary the use of identity management in service providers, preventing users from handing out their credentials in different Web sites (implementing Single Sign-On) and giving them the opportunity to increase the control over their user profile managing the information provided to the service provider.

The aim of this project is the provision of an electronic commerce service that utilizes a next-generation network for authorization purposes. Taking advantage of development opportunities they offer, and developing a module for authentication and verification of financial transactions through the network by using a mobile phone.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1 Motivación del proyecto

De manera simultánea al crecimiento experimentado por Internet, lo hace el número de sectores con presencia en dicha red [1]. Uno de los sectores con una mayor proyección es el comercio electrónico, que goza de una gran popularidad y cuyo desarrollo puede ser de vital importancia en los próximos años [2].

El despliegue del comercio electrónico se ha encontrado limitado por diferentes factores, como la falta de confianza de los usuarios a comprometer su información privada por la red o la falta de estandarización para la realización de pagos presente en el sector. Sin embargo, la aparición de un nuevo abanico de soluciones de pago que incluyan la protección de la identidad del cliente y la gestión de las operaciones a través de plataformas seguras e interoperables pueden lograr que el comercio electrónico sea cada vez una realidad más fuerte.

En las últimas décadas, el pago a través de Internet se ha realizado mediante el uso de tarjetas de crédito, debido a su gran aceptación y facilidad de su uso. Como sistema de pago físico, las tarjetas de crédito requieren la verificación por parte del dependiente de las credenciales (típicamente el DNI o un PIN) del cliente. Sin embargo, en escenarios “on-line” la figura del dependiente no existe, y debido a la inexistencia de una identidad digital que permita verificar que la tarjeta pertenece al cliente, la aplicación de dicho sistema de pago en las transacciones electrónicas presente un bajo nivel de seguridad y confianza.

La gestión de la seguridad es un requisito indispensable en las transacciones económicas. Por ello, la aparición de tarjetas EMV (Europay MasterCard Visa) [3] en sustitución de las tarjetas de banda, ha supuesto una mayor protección frente al fraude. Para ello, dichas tarjetas incorporan algoritmos de cifrado como DES, Triple DES o RSA, que permiten a la tarjeta y al lector de la misma autenticarse mutuamente y respecto a la entidad que se ocupa de la transacción.

Por otro lado, el estándar EMV establece una interoperabilidad segura entre tarjetas y terminales que cumplan EMV, restringiendo su uso exclusivamente a los sistemas adaptados para la lectura de tarjetas EMV, no permitiendo el uso de los lectores convencionales de tarjetas de banda magnética existentes en el mercado. Su uso, consecuentemente, se encuentra en la actualidad reducido al comercio convencional, debido al gran coste asociado la adaptación de los terminales de los usuarios para el uso de las tarjetas EMV para la compra a través de la red.

Con el fin de proporcionar la seguridad necesaria a las transacciones realizadas en Internet surgen las denominadas pasarelas de pago [4], que actúan como entidades que

facilitan la transferencia de información entre el servidor web del vendedor y la entidad bancaria a la que está asociado el comprador de manera transparente al mismo.

El comercio a través de Internet cuenta con una gran variedad de fuentes de amenaza, entre las que se encuentran el uso de spyware, la suplantación de sitios web o ataques como el phising o el pharming. Los ataques anteriormente comentados dotan en la actualidad de una gran sofisticación y evolucionan de manera simultánea a los sistemas a los que amenazan, ocasionando grandes pérdidas en los mismos [5].

La motivación de este proyecto es el diseño de una pasarela de pago a través de varios canales de comunicación (Internet y telefonía móvil), introduciendo el uso de terminales IMS para la verificación de las transacciones económicas. La mayor robustez de la telefonía móvil a la suplantación de identidad y a los diferentes ataques mostrados anteriormente, junto al uso de tecnologías de gestión de identidad para la provisión de privacidad a los usuarios, proporcionan al sistema un valor añadido frente al pago que tradicionalmente se emplea en la red y que consiste en proporcionar el número de tarjeta. A su vez, la mayor dificultad de comprometer las redes de telefonía móvil e Internet de manera simultánea, hace que el mecanismo de pago desarrollado en el proyecto sea muy robusto frente ataques del exterior.

La existencia de canales de comunicación diferenciados será en un futuro no muy lejano sustituido por un modelo de red convergente [6], denominado red de próxima generación, en el que los servicios puedan ser accedidos desde puntos de acceso tan dispares como un teléfono móvil, un ordenador o una televisión. Por ello, el desarrollo de la pasarela de pago se ha realizado orientado a su uso en una red IMS [7], ya que es necesaria la adaptación de servicios como el comercio electrónico a las oportunidades de gestión que el nuevo modelo de red nos ofrece, generando una posible respuesta a los requisitos en materia de gestión de identidad y de seguridad demandadas por el mercado.

1.2 Objetivos

El principal objetivo de este proyecto es crear una pasarela de pago que permita la realización de transacciones económicas a través de una red multicanal IMS, protegiendo la identidad del usuario y dotando al sistema de un alto grado de confianza al soportar autenticación a dos niveles y por dos canales de comunicación diferentes.

A su vez, se pretende desarrollar un servicio tan común como el comercio electrónico, como caso de uso de la red IMS, a modo de estimar las grandes posibilidades que dicho modelo de red ofrece para tal fin.

Una enumeración más detallada de los objetivos perseguidos por el proyecto se proporciona a continuación:

- Realizar un estudio de las herramientas de código abierto existentes para la gestión de identidad.

- Desarrollar un sistema para gestión de identidad, formado por un proveedor de servicios, un proveedor de identidad y varios módulos de autenticación. Se utilizan varios módulos de autenticación debido a que el sistema soportará niveles de autenticación diferenciados según la acción que el usuario desee realizar.

- Probar el correcto funcionamiento del sistema de gestión de identidad y evaluar la interoperabilidad de los diferentes módulos de autenticación presentes en el mismo.

- Desarrollar un sistema de pasarela de pago que reciba la información del proveedor de servicios y acceda a los proveedores de identidad para proporcionar un servicio transparente al usuario.

- Desplegar un proveedor de servicio sencillo y fácil de utilizar por el usuario para comprobar el funcionamiento del sistema.

- Desarrollar una aplicación cliente para comprobar el correcto funcionamiento de la autenticación por terminal móvil, que consistirá en un plugin para el soporte de mensajería instantánea para confirmación de compras en un terminal IMS.

- Comprobar el correcto acceso al proveedor de identidad desde el dispositivo IMS.

- Dotar al sistema de protección frente a la suplantación de identidad mediante el uso de firmas digitales en los mensajes intercambiados entre proveedor de identidad y terminal IMS del cliente.

1.3 Contenido de la memoria

La estructura, en cuanto a capítulos se refiere, de la memoria es el siguiente:

- En el Capítulo 2, “Estado del arte”, se introduce el concepto de Red de Próxima Generación o NGN (Next Generation Network, en inglés), mostrando las principales características del modelo de red convergente.

  Por otro lado, se especifica la situación actual del comercio electrónico en Internet, analizando los puntos fuertes y las vulnerabilidades o limitaciones existentes para cada una de las soluciones del mercado. También se realiza un estudio acerca de las herramientas existentes para la gestión de la identidad y los requisitos básicos en materia de seguridad para las aplicaciones que operan en la red.

- En el Capítulo 3, “Descripción general del sistema”, realiza una descripción general del sistema desarrollado, desde sus requisitos funcionales a la arquitectura que presenta, detallando los diferentes módulos que componen el sistema completo. Se incluyen diagramas para clarificar el funcionamiento y la interconexión de los módulos.

- En el Capítulo 4, “Módulo de gestión de identidad” se describe más detalladamente el sistema de gestión de identidad, formado por un proveedor de identidad y la

aplicación de comercio electrónico. Ambos serán analizados en profundidad así como las herramientas utilizadas para su configuración y desarrollo.

- En el Capítulo 5, “Módulo de verificación de pago seguro a través de terminal IMS”, se describe el módulo de verificación de los pagos, formado por la pasarela de pago, el módulo de autenticación a través de la red IMS y la aplicación cliente.

  Se detallarán las configuraciones e implementaciones realizadas para el despliegue de dicho módulo, así como su interoperabilidad con el módulo de gestión de identidad mencionado anteriormente.

- En el Capítulo 6, “Pruebas”, se detallan tanto el diseño como los resultados obtenidos de las pruebas realizadas sobre el sistema completo, que han sido orientadas a comprobar la comunicación entre módulos de autenticación, pasarela de pago, proveedor de servicios y cliente, así como el correcto funcionamiento del sistema de gestión de identidad, ya que el funcionamiento de cada uno de los módulos está directamente relacionado al del resto de ellos.

- En el Capítulo 7, “Historia del proyecto”, se detallan las diferentes fases de desarrollo del proyecto, incluyendo las tareas realizadas y las dificultades encontradas en cada una de ellas.

- En el Capítulo 8, “Conclusiones”, se establecen las conclusiones sobre el desarrollo del presente proyecto, así como se realiza un análisis de posibles líneas de trabajo futuras en el campo que ocupa este texto.

- En el Apéndice A, “Presupuesto”, se detalla el presupuesto asociado a la realización del presente proyecto, teniendo en cuenta tanto los costes asociados al personal empleado en el desarrollo del mismo, como los costes del material empleado en dicho desarrollo.

- En el Apéndice B, “Manual de instalación”, se detallan los pasos a seguir para instalar tanto el módulo de autenticación sobre OpenSSO desarrollado como del plugin de mensajería asociado a la aplicación cliente.

- En el Apéndice C, “Glosario de términos”, se muestra un glosario que recoge los principales términos y conceptos asociados al proyecto.

Capítulo 2

Estado del arte

2.1. Introducción

La protección de datos de clientes y la seguridad de los procesos de compra son los dos pilares fundamentales a los que se orienta la investigación sobre las tecnologías para el pago electrónico. El creciente uso de Internet para la realización de transacciones económicas conlleva la necesidad de adaptar los métodos de pago tradicionales para su uso en la Web.

La falta de seguridad en la red y la actual inestabilidad de los sistemas de pago electrónico en cuanto a estandarización, hacen que el abanico de soluciones no esté cerrado. Por ello, se plantea que nuevas posibilidades aparezcan con la solución a los problemas anteriormente expuestos, estableciendo mecanismos de pago seguros e interoperables.

La gestión de la identidad juega un papel clave a la hora de integrar la identidad digital del usuario en diferentes servicios permitiéndole gestionar sus datos de forma sencilla y evitar abusos o comprometer dichos datos. El uso de identidad federada y de identidad "user-centric" (centrada en el usuario) permite a los usuarios acceder a recursos de otros dominios de forma segura utilizando una o varias identidades digitales. Actualmente existen numerosas iniciativas, entre las que se encuentran OpenID, SAML o SXÎP, cuyo papel es esencial en aplicaciones en las que la protección de la identidad del usuario y la verificación de la autenticidad de ciertos atributos de la misma son cruciales.

El uso de mecanismos de firma digital es otro de los temas abordados en el proyecto, debido a la necesidad de proteger los datos en un entorno en el que se produce intercambio de documentos con información privada de los usuarios.

Por otro lado, es necesario introducir el concepto de red de próxima generación, como es el caso de IMS, y el de aplicaciones convergentes que pueden desarrollarse para llevar información desde la Web a redes móviles, haciendo posible que la comunicación se produzca a través de diferentes canales de manera conjunta.

2.2. El comercio electrónico

Desde hace varias décadas el área del comercio electrónico ha experimentado un importante crecimiento, originado principalmente por la aparición de Internet y sus tecnologías asociadas. El gran desarrollo experimentado por las TIC (Tecnologías de la Información y la Comunicación), ha influido en la forma de hacer negocios, denotando la importancia que juegan los métodos de pago electrónico para que los negocios digitales tengan un éxito real en el futuro.

El comercio electrónico se define como el área que permite a los usuarios comprar productos, obtener servicios e intercambiar información de transacción de negocios "on-line". El principal canal que ha desarrollado el comercio electrónico ha sido Internet, ya que su

impacto ha sido mucho mayor que el de otros canales existentes. Entre ellos se encuentran canales tales como el intercambio electrónico de datos o EDI (Electronic Data Interchange en inglés) a través de VAN (Value Added Networks) [8][8], que ofrecen una disponibilidad absoluta de los datos aunque no permiten la capacidad de análisis y de acción sobre los mismos que Internet proporciona.

Con la aparición de Internet, el comercio electrónico tuvo una gran aceptación, pasando del 2,5 % de representación de páginas comerciales en la red en 1993, a un 50% tan sólo dos años más tarde. Sin embargo, también se pudo comprobar, y sobre todo en el comercio B2C, que la red tan sólo ofrecía un entorno ventajoso para ciertos tipos de productos y servicios determinados. Según dicha apreciación, serían, los productos con mayor coste, diferenciación e intangibilidad que pudieran ser pedidos y pagados por la red, los que gozarían de un mayor futuro de éxito.

Atendiendo a los objetivos de negocio que persiga, existen diferentes tipos de comercio electrónico:

- Business to Consumer (B2C): Comercio electrónico en el que no existen intermediarios y los consumidores compran los productos directamente a la empresa.

- Business to Business (B2B): Comercio electrónico automatizado en que varias empresas compran y venden sus productos entre ellas. La información de los productos intercambiados está disponible para las partes implicadas y es modificable en tiempo real.

Este texto se centrará en el comercio electrónico B2C, realizando un análisis de los diferentes mecanismos de pago existentes en la actualidad, las herramientas utilizadas en cada uno de ellos y las ventajas e inconvenientes que presentan para satisfacer las necesidades de los usuarios y del mercado del comercio electrónico.

2.2.1. Métodos de pago tradicionales

El propio desarrollo del comercio electrónico está demostrando que su mayor fuente de beneficios se encuentra en escenarios en los que todas las operaciones que forman parte de las transacciones (pedido, facturación, cargo…) se realizan de forma electrónica.

Uno de los métodos de pago electrónico tradicionales más utilizados son los cheques electrónicos y las transferencias electrónicas. El uso de estos mecanismos de pago exige la intervención de las entidades bancarias de cada uno de los participantes en la transacción económica realizada, que entran en contacto una vez se les comunique por parte de sus clientes. La única restricción es que las entidades bancarias se mantengan fuera de contacto del cliente del otro banco, denominada separación de agentes bancarios o modelo de China Wall [9].

La Figura 2.1 muestra el esquema del método de pago tradicional y los agentes involucrados en el mismo, que son:

- Cliente: Persona que realiza la compra en el comercio.

- Comercio: Establecimiento en el que el cliente realiza la compra.
- Banco del cliente: Entidad financiera asociada al cliente.
- Banco del comercio: Entidad financiera asociada al comercio.

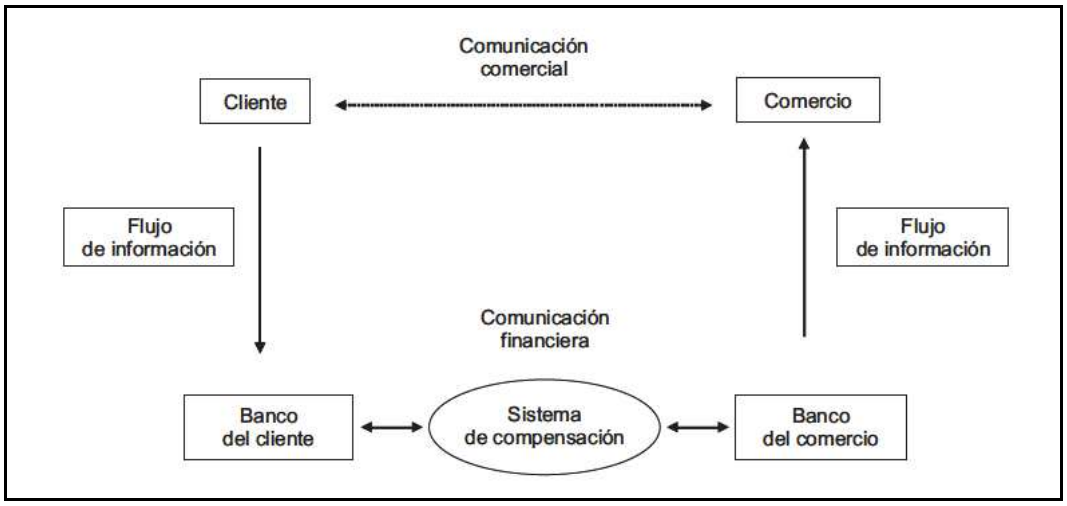

Figura 2.1: Esquema de sistema de pago tradicional. Obtenida de [8]

Por ello, los métodos de pago tradicionales, como el pago en efectivo contra reembolso, con cheque o por transferencia bancaria, a pesar de contar con un elevado grado de confianza del usuario, quedan fuera de uso en las transacciones de tipo B2C. La rapidez, la comodidad y el ahorro de tiempo son las principales variables a tener en cuenta en dicho entorno, relegando el uso de estos métodos de pago tradicionales en actualidad a las transacciones de tipo B2B.

En el caso de los métodos de pago tradicionales, la ventaja derivada de su uso queda clara:

- Elevado grado de confianza en las entidades bancarias y en las transacciones realizadas directamente en sus oficinas.

Sin embargo, las desventajas para su uso en el comercio electrónico son varias:

- Poca fluidez en las operaciones.
- Necesidad de desplazarse a la oficina de la entidad bancaria para concretar las transacciones.
- Imposibilidad de compra inmediata, ya que la liquidación de la transacción se realiza a posteriori del pedido del producto.

2.2.2. Pago con tarjeta

El método de pago por tarjeta engloba al pago con tarjetas de crédito y de débito, y requiere la afiliación del usuario a una entidad bancaria mediante la apertura de una cuenta.

En cuanto a funcionamiento, el modelo de pago con tarjeta tiene algunas características diferentes al método tradicional, ya que en su caso el proveedor transmite la información a su banco, y éste a su vez al sistema de compensación (Visa, MasterCard). Este método de pago necesita a su vez de la previa autorización del banco emisor de la tarjeta para que se produzca la transacción de manera satisfactoria, pudiendo ser dicha autorización on-line o diferida.

La autorización de la tarjeta proporciona seguridad al comercio y favorece al intercambio de información entre el operador del sistema y las instituciones financieras. Este hecho favorece la creación de nuevos elementos en el sistema, como el operador del sistema (Visa Master Card o American Express) y la gestión de información entre bancos.

La Figura 2.2 muestra el esquema del método de pago con tarjeta y los agentes involucrados en el mismo, que serán:

- Cliente: Persona que realiza la compra en el comercio.
- Comercio: Establecimiento en el que el cliente efectúa la compra.
- Banco del emisor: Entidad financiera asociada al cliente o emisor del pago.
- Banco del Merchant: Entidad financiera asociada al comercio en el que se realiza el pago.
- Operador del sistema: Entidad que gestiona las transacciones realizadas entre las entidades financieras asociadas al cliente y al comercio, respectivamente.

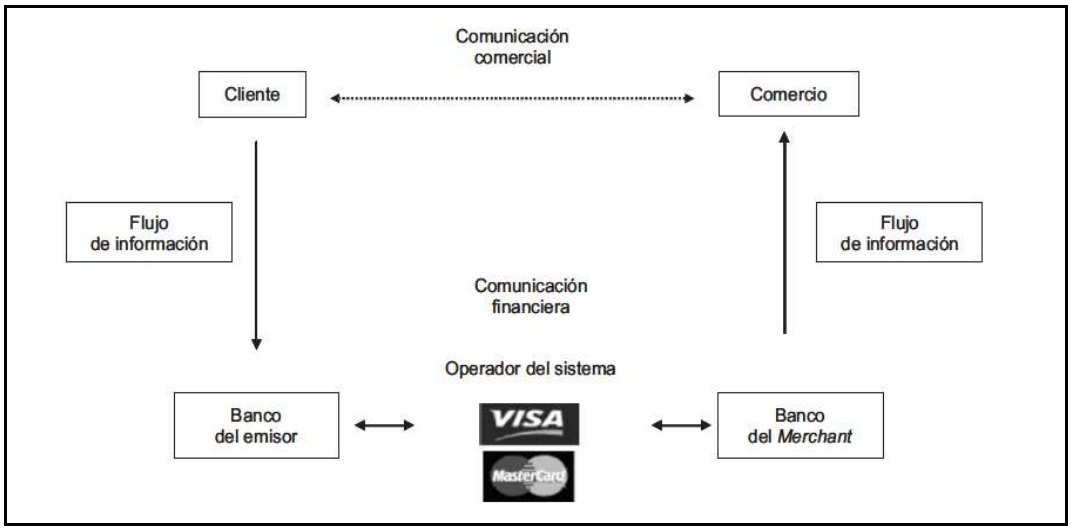

Figura 2.2: Esquema de sistema de pago con tarjeta. Obtenida de [8]

Sin embargo, el uso de las tarjetas de crédito sigue acarreando la falta de confianza por gran parte de los usuarios del comercio electrónico, por lo que se han estudiado diferentes soluciones para proveer al pago con tarjeta de un mayor nivel de seguridad.

A continuación vamos a detallar algunos de los principales problemas existentes en la transferencia de información en Internet, que se extienden a los pagos realizados a través de dicha red y que deben ser solucionados para aumentar el grado de confianza del usuario y no frenar así el crecimiento del sector:

- Confidencialidad: Trata que la información no sea accedida por personas no autorizadas a tal efecto.
- Privacidad: Trata de proteger a los sistemas de comunicación frente al robo de información.
- Validación de la identidad o autenticación: Trata de determinar con total fiabilidad la identidad real de la persona con la que se va a producir el intercambio de información, antes de que éste se produzca.
- Irrefutabilidad o No repudio: Trata de evitar la negación por parte de alguna de las partes implicadas en el intercambio de información de su participación en la misma.
- Integridad: Trata de asegurar que la información trasmitida a través de una red de ordenadores no sea modificada en su camino a través de ella.

Todos los puntos anteriores, y especialmente los tres últimos, resultan claves a la hora de realizar transacciones B2C con tarjetas de crédito. El problema real reside en la poca disposición de los usuarios a proporcionar su información de identidad y la de sus tarjetas, debido al elevado riesgo de su uso en la red.

La conclusión es que las tarjetas de crédito no han conseguido ganarse la confianza de sus usuarios como mecanismo de pago electrónico en cuanto a la fiabilidad y seguridad debido a la carencia real de seguridad de las mismas, además del factor psicológico en contra de su uso generado por dicha desconfianza.

Como consecuencia de ello, la solución de los problemas de seguridad derivados del uso de tarjetas de banda magnética en comercios electrónicos, está íntimamente relacionada con el desarrollo de sistemas de gestión de identidad y de autenticación más robustos que los existentes, y que permitan al usuario confiar en el sistema al que proporcionan sus datos.

A continuación vamos a estudiar diferentes tecnologías orientadas a reforzar la seguridad en las transacciones económicas con tarjetas de crédito y así aumentar su grado de aceptación por el usuario.

2.2.2.1. SSL (Security Sockets Layer)

Es el protocolo de seguridad más común en la red. Diseñado por Netscape Communications Inc., establece un nivel de transporte seguro entre el servicio de transporte en Internet (TCP) y las aplicaciones que se comunican a través del mismo. Más adelante la IETF lo estandarizó como TLS (Transport Layer Security) incluyendo varias mejoras.

La comunicación entre cliente y servidor se produce de la siguiente manera:

- Se negocia entre ambas partes el mecanismo a utilizar en la comunicación, típicamente mediante el uso de un certificado de clave pública tipo X.509 (PKI, Public Key Infrastructure) [10].

- El cliente genera una clave simétrica de sesión y la envía cifrada con la clave pública del servidor.

- Una vez conocida por ambos la clave de sesión, se intercambian información cifrada con la clave secreta.

El sistema usa un mecanismo de claves públicas generadas por diferentes Autoridades de Certificación (CA en inglés, Certificate Authority), y permite a las aplicaciones cliente-servidor comunicarse proporcionando confidencialidad, integridad y autenticación. De esta manera, se evitan las escuchas (eavesdroping), la falsificación de identidades (phishing) y se asegura la integridad de los mensajes.

SSL establece un canal seguro para la realización de transacciones no excesivamente complejas, pero sin embargo carece de capacidad para completar el proceso comercial: verificar la validez del número de tarjeta recibido, autorizar la transacción con el banco del cliente o procesar el resto de la operación con los bancos adquiriente y emisor.

Además, SSL sólo garantiza confidencialidad e integridad de los datos en tránsito, y no en origen y en destino, lo que debilita la seguridad de la información de las tarjetas de los usuarios almacenadas en los comercios electrónicos y reduce la eficiencia de su utilización en la realización de pagos electrónicos.

2.2.2.2. SET (Secure Electronic Transport)

Como complemento al protocolo SSL, MasterCard y Visa desarrollaron SEPP (Secure Electronic Payment Protocol) y STT (Secure Transaction Technology), para asegurar las transacciones económicas mediante el uso de tarjetas de crédito. Posteriormente, y con la ayuda de American Express, desarrollaron un protocolo único para dicho objetivo, denominado SET [11].

El protocolo SET es un conjunto de especificaciones y normas de seguridad, que definen un estándar para la realización de transacciones económicas en Internet, cuyos principales objetivos son:

- Proteger el sistema de tarjetas de crédito para su uso en Internet.

- Promover la confianza en las transacciones en la red entre los usuarios.

- Promover el desarrollo y la definición de nuevos mecanismos de pago seguros e interoperables.

Su implantación, por otro lado, supone una serie de beneficios inmediatos:

- Autenticación de todas las entidades participantes en las transacciones comerciales: titulares de las tarjetas, comercios y bancos.

- Garantía de la confidencialidad del pago.
- Asegurar la integridad de los mensajes financieros, no siendo manipulados en el denominado circuito de pago.
- Proporciona interoperabilidad entre distintas plataformas hardware y software.
- Evita el pago mediante tarjetas no autorizadas.
- Evita el robo de información financiera del comprador.
- Posibilidad de mejorar el grado de confianza de los usuarios, al proporcionar información de seguridad a los mismos, al marcar las transacciones con la etiqueta "SET COMPLIANT".

Finalmente, cabe reseñar que el uso del protocolo SET garantiza el cumplimiento de las tres principales condiciones establecidas en [12]:

- Confidencialidad: La información que es transferida por la red se encuentra protegida por métodos criptográficos que cifran los datos intercambiados entre las partes implicadas.
- Autenticación: Se realiza a través de certificados digitales que poseen comprador y vendedor, proporcionados por una tercera parte, la entidad financiera, por ejemplo, VISA. Dichos certificados sustituyen la función realizada por las tarjetas de crédito convencionales.
- Integridad: El uso de firmas digitales hace que la integridad de los datos quede asegurada junto con la autenticidad de los mismos.

Pese a lo indicado anteriormente, el uso de SET no es generalizado y, a pesar de ser un protocolo más seguro que SSL para pagos electrónicos, su implementación supone un mayor tiempo de procesamiento, y como consecuencia, unos mayores costes de aplicación y mantenimiento, por lo que su despliegue parece poco probable.

2.2.2.3. Pasarelas de pago

Una pasarela de pago [13], también conocida como "Gateway" de pago, es una aplicación que autoriza pagos a negocios electrónicos. Las pasarelas cifran información sensible como los números de las tarjetas de crédito, garantizando que la transferencia de información entre cliente y servidor se efectúa de manera segura.

El papel desempeñado por la pasarela de pago es el de intermediario entre el sitio web de comercio electrónico y la entidad financiera que recibe el pago. A continuación se describen los principales participantes en el procesamiento de una transacción electrónica mediante una pasarela de pago:

- Usuario: Es el cliente que realiza la compra en un comercio electrónico a través de su navegador conectado a Internet.

- Comercio electrónico: Es el establecimiento en el que el usuario realiza la compra.
- Banco emisor: Es la entidad financiera asociada al usuario o cliente y que recibe la orden de pago.
- Banco adquiriente: Es la entidad financiera asociada al comercio electrónico y que recibe la confirmación del pago.
- TPV Virtual: Es el terminal punto de venta que actúa como intermediario entre las entidades bancarias asociadas al usuario y al comercio electrónico, gestionando las órdenes de pago y las confirmaciones de los mismos.

Al ordenarse el pago por parte de un cliente en un comercio que tenga habilitado el servicio de pasarela de pago, se realizan una serie de procesos transparentes al comprador:

- El cliente confirma la compra pulsando el botón pagar o similar e introduce los datos de su tarjeta de crédito. Los datos requeridos a introducir pueden ser:
  - Nombre completo del titular.
  - Número de la tarjeta.
  - Fecha de caducidad.
  - Código CSC (Card Security Code) [Figura 2.3]

Figura 2.3: Código CSC asociado a las tarjetas de crédito/débito

- El navegador Web del cliente cifra mediante SSL la información hacia el servidor Web del vendedor.
- El servidor del vendedor reenvía los detalles de la transacción al servidor de pago situado en su pasarela de pago, que posee la información de las cuentas de los comercios.
- La pasarela de pago reenvía la información a la entidad bancaria del vendedor.
- El banco emisor de la tarjeta recibe un pedido de autorización y envía la respuesta a la pasarela de pago, a través de la entidad bancaria del vendedor, con un código de respuesta, que servirá para determinar si se aceptó o se rechazó la operación y, en el segundo de los casos, por qué motivo ocurrió.
- La pasarela de pago recibe la respuesta y la reenvía a la Web que procesa el pago, donde se informará al cliente del estado de la transacción.

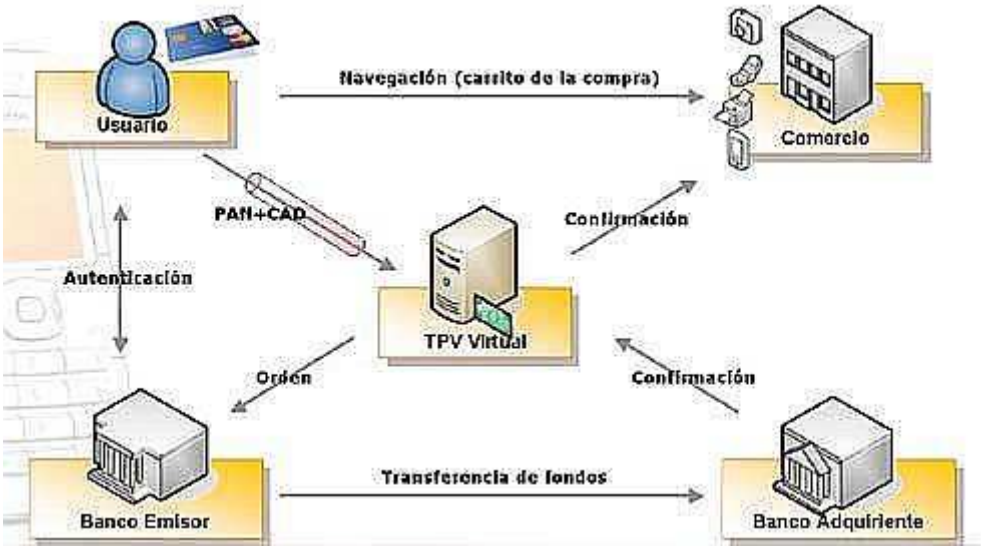

Figura 2.4: Esquema del sistema de validación de compras

El proceso completo de validación de la compra puede durar varios segundos. Las cuentas bancarias de comprador y vendedor pueden pertenecer a la misma entidad financiera o a entidades diferentes.

El uso de pasarelas de pago proporciona un valor añadido en materia de seguridad de las transacciones y de la protección de datos de identidad de los usuarios, pero aún en la actualidad surgen nuevas amenazas que hacen peligrar la estabilidad del modelo descrito, ya que no se protege frente a ataques como el phising, spyware o pharming.

A modo de resumen, vamos a detallar las ventajas e inconvenientes de las pasarelas de pago en su uso en las transacciones económicas:

Ventajas:

- Protección de los datos bancarios del cliente, ya que su información se la da a la pasarela y no al vendedor.
- El cliente puede verificar la cantidad exacta que se le va a cargar antes de introducir los datos de su tarjeta de crédito.
- La información sensible, como el número de tarjeta de crédito usado en una compra, es cifrada y viaja segura teniendo como único destino el servidor de la pasarela.
- El cliente puede seleccionar entra varias tarjetas de crédito para efectuar el pago.
- Se verifica que la tarjeta de crédito es real, e incluso si posee fondos suficientes para efectuar el cobro.

- El sistema de cobro es válido para clientes de todo el mundo debido a que las entidades financieras poseen una relación de confianza entre ellas.

Inconvenientes:

- Amenazas propias de la red que pueden afectar a las pasarelas de pago, descritas a continuación, dado que dichas pasarelas no se protegen contra ellas:

  - Phishing: Delito encuadrado dentro de las estafas cibernéticas que consiste en la suplantación de la identidad de una persona o una empresa, mediante una comunicación oficial electrónica aparentemente válida, para sustraer información privada del usuario.

  - Spyware: Programa encuadrado dentro del Malware, que se instala furtivamente en un ordenador para recopilar información acerca de las actividades realizadas en el mismo y la información privada almacenada y enviada por él.

  - Pharming: Explotación de la vulnerabilidad en el software de los servidores DNS, o el fichero "hosts" de un ordenador de forma que las peticiones HTTP a páginas legítimas se redirijan a páginas establecidas por el atacante.

2.2.3. Monederos electrónicos

Los monederos electrónicos son tarjetas prepago provistas de un microchip donde almacenan una cantidad que corresponde a dinero. Su uso está orientado a micropagos en lugares (pequeños comercios, taxis, cabinas telefónicas, locutorios) donde el uso de tarjeta de crédito no estaba permitido y era necesario el pago en efectivo.

La carga de dinero en la tarjeta se realiza mediante una transferencia bancaria desde la cuenta del usuario de la tarjeta o a través de una tarjeta de crédito o débito asociada al monedero. Una vez actualizado el saldo del mismo, puede ser utilizado para comprar en cualquiera de los establecimientos adheridos a este método de pago.

Para la validación de la tarjeta, ésta debe ser introducida en el dispositivo lector que el vendedor tendrá instalado en su establecimiento, quedando la operación realizada registrada y el saldo actualizado en la tarjeta. El importe cargado se registra tanto en la propia tarjeta (a través del microchip incorporado en la misma), como en los archivos del servidor del banco.

El uso fraudulento o irregular de las tarjetas debido a su pérdida o sustracción corre bajo responsabilidad del propio titular de las mismas, incluso después de haber notificado a la entidad bancaria correspondiente. El motivo es que podrían considerarse como dinero en efectivo.

En España, los monederos cuyo uso se encuentra más extendido son:

- Monedero 4B [4].

- Monedero Euro 6000 [14].

- Monedero VisaCASH.

Uno de los problemas surgidos del uso de dichos monederos es la falta de interoperabilidad entre ellos, haciendo que la compra con un monedero determinado sólo pueda realizarse en un establecimiento adherido al mismo.

Basándose en este hecho, se han realizado grandes esfuerzos por la creación de especificaciones para la definición de monederos electrónicos interoperables, como EMV (Europay Mastercard Visa) [3] o CEPS (Common Electronic Purse Specification) [15] .

2.2.3.1. EMV

Estándar de tarjeta inteligente (con microchip integrado) creado por Europay, MasterCard y VISA, de ahí sus siglas, para establecer unas especificaciones comunes en cuanto a aplicaciones, terminales y tarjetas para métodos de pago.

EMV define la interacción de las tarjetas IC ("Integrated Circuit Cards") y los dispositivos lectores de las mismas a nivel físico, eléctrico, de datos y aplicación, para transacciones finales.

Parte del estándar se basa en el concepto "IC Chip Card" y existe compatibilidad con las tarjetas "Carte Bleue", desplegadas en Francia, aunque actualmente éstas se encuentran migrando hacia EMV.

A pesar de que existen variedad de implementaciones, la mayoría de los terminales EMV comprueban la identidad del propietario de la tarjeta a través del tecleo de un código PIN, sustituyendo a la firma en papel. La autenticación mediante PIN depende de las posibilidades del terminal y la programación de las tarjetas.

Las transacciones financieras basadas en EMV ofrecen una mayor protección frente al fraude que las tarjetas de banda magnética tradicionales. Esto se debe al uso de algoritmos de cifrado más potentes tales como DES, triple-DES o RSA, que permiten a la tarjeta y al lector (o terminal de pago) autenticarse mutuamente así como al centro de procesamiento de la transacción.

Además, el aumento de interoperabilidad entre monederos electrónicos supone un factor clave a la hora de convencer al usuario para realizar sus pagos mediante este método. EMV intenta solucionar estos problemas y hacer posible la estandarización de los mecanismos de compra en todo el mundo.

Otra de las ventajas que presentan las tarjetas EMV, es la posibilidad de implementar nuevos mecanismos de seguridad, como la autenticación biométrica, sustituyendo al uso de código PIN y proteger así al sistema frente a ataques de suplantación de identidad

Sin embargo, el tiempo de procesamiento requerido por las tarjetas EMV es sensiblemente mayor que el de las tarjetas de banda magnética, debido al coste computacional necesario por el uso de criptografía. Además, su uso en operaciones a través de

Internet se ha visto limitado por la necesidad de un hardware específico (lector certificado) validado por EMVCo para la lectura de las tarjetas que debería estar en posesión de los usuarios.

2.2.4. Pago con teléfono móvil

El auge del comercio electrónico, seguido de los problemas de inseguridad respecto al uso de tarjetas de crédito en la red, ha ocasionado la aparición de iniciativas que promueven el uso del teléfono móvil como herramienta de pago.

El uso de los terminales móviles se considera ya como una alternativa real al uso de dichas tarjetas, ya que los usuarios aún no han superado la desconfianza que supone introducir los datos de sus tarjetas a través de Internet.

La mayoría de población no afiliada a una entidad bancaria, sobre todo la gente más joven, suele tener acceso a un teléfono móvil. Por esta razón, dichos terminales resultan atractivos como forma de pago.

El uso del teléfono móvil, como medio de identificación y pago, ofrece alta portabilidad, seguridad, penetración, conectividad y personalización. Además, el coste para el usuario y el nivel de autenticación requerido se minimizan, ya que en algunos sistemas no es necesario siquiera proporcionar el número del terminal para identificarse, ya que la información del usuario se encuentra almacenada en la tarjeta SIM del terminal.

El pago mediante teléfono móvil soporta diversas modalidades, como son el prepago, el ingreso mediante tarjeta de crédito o cuenta corriente o la facturación junto con los costes del teléfono móvil. Dichas opciones difieren en los requisitos necesarios para realizar el pago y en las tecnologías utilizadas en el mismo.

Ventajas del pago con teléfono móvil:

- Autenticación segura del usuario minimizando el riesgo de suplantaciones, ya que en caso de pérdida o sustracción del teléfono habría que cancelar la tarjeta SIM asociada al mismo.
- Aplicable a pequeñas compras (micropagos).
- Comodidad de uso. Pago rápido y sencillo.
- Realizar compras sin necesidad de facilitar datos personales.
- Alternativa en situaciones donde se requiere la movilidad.

Inconvenientes del pago con teléfono móvil:

- Protección legal del consumidor mediante la Ley Orgánica de Protección de Datos (LOPD).
- Necesidad de aceptación como medio de pago fiable.

- Ausencia de estándares para este método de pago.
- Grandes inversiones necesarias en los comerciantes, ya que necesitaría disponer del equipamiento y la infraestructura necesaria para aceptar el pago de los diferentes operadores.

El teléfono móvil aún no se encuentra asentado definitivamente como medio de pago, siendo necesario motivar tanto a comerciantes como a usuarios para su uso. Para ello se debería incidir en sus puntos fuertes, como por ejemplo la posibilidad de pagar en un comercio desde cualquier punto del mismo y no sólo en la caja.

Roles y componentes de un sistema de pago con móvil:

En la Figura 2.5 se muestran los diferentes roles necesarios en una transacción financiera a través del móvil, representados como rectángulos, y los componentes o funciones utilizados en dichas aplicaciones, representados como óvalos.

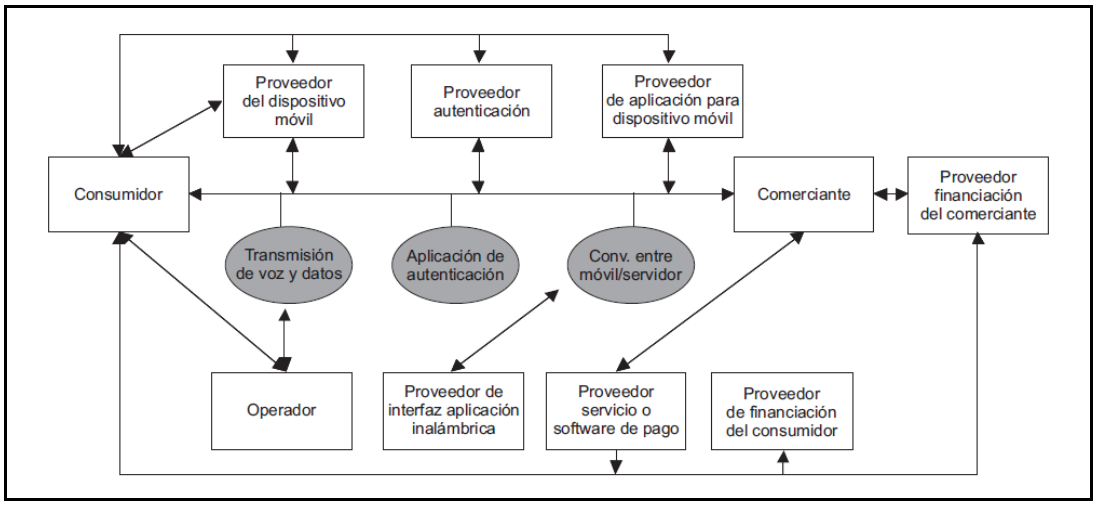

Figura 2.5: Roles involucrados en las transacciones económicas a través de telefonía móvil. Obtenida de [8].

Existen nueve roles básicos en las transacciones por móvil, pudiendo ser varios de ellos desempeñados por la misma compañía:

- Consumidor.
- Comerciante o proveedor de servicios.
- Fabricante del dispositivo móvil.
- Proveedor de interfaz de pago inalámbrico (separa la información de pago del resto hasta el proveedor del sistema de pago).

- Proveedor de servicios de autenticación del consumidor.
- Proveedor de la aplicación en el dispositivo móvil para realizar la transacción.
- Proveedor del software para la ejecución de la transacción.
- Proveedor de financiación del comerciante.
- Proveedor de financiación del consumidor.

Los tres componentes funcionales básicos de este tipo de transacciones son:

- Transmisión de los datos contenidos en el mensaje de pago.
- Autenticación del comprador o consumidor para evitar el fraude.
- Conversión de datos de pago entre el móvil y el servidor que procesa la transacción.

Las relaciones, componentes y roles participantes en el modelo de pago son interdependientes, y cada uno necesita del siguiente para realizar la transacción con éxito. El proceso de estandarización de todos estos roles y componentes, y la coordinación entre las partes implicadas (entidades financieras, operadoras telefónicos y sistemas de compensación), suponen las principales barreras al despegue del pago con móvil.

2.2.4.1. Valimo

Valimo [16] es una empresa finlandesa de telecomunicaciones que ha desarrollado un dispositivo móvil para autenticar al usuario digitalmente, realizar firmas digitales sobre documentos y confirmar transacciones o pagos. Mediante la simple introducción de un código PIN da acceso a servicios tan dispares como banca on-line, pagos móviles, comercios electrónicos, identidad empresarial o control de acceso.

Valimo desempeña varios de los roles descritos con anterioridad, tales como proveedor de interfaz de pago inalámbrica, de servicios de autenticación del consumidor y de la aplicación en el dispositivo móvil para realizar la transacción.

Valimo permite el uso de servicios que requieran validación de atributos del usuario. Una vez que el usuario introduzca su código PIN, se valida la transacción de atributos mediante criptografía y el servicio solicita confirmación para efectuar las acciones correspondientes. En el caso concreto de la compra en un comercio electrónico, le enviará al terminal Valimo la orden de confirmar el pago. Una vez el usuario pulse la tecla correspondiente, dicho pago se habrá realizado con éxito.

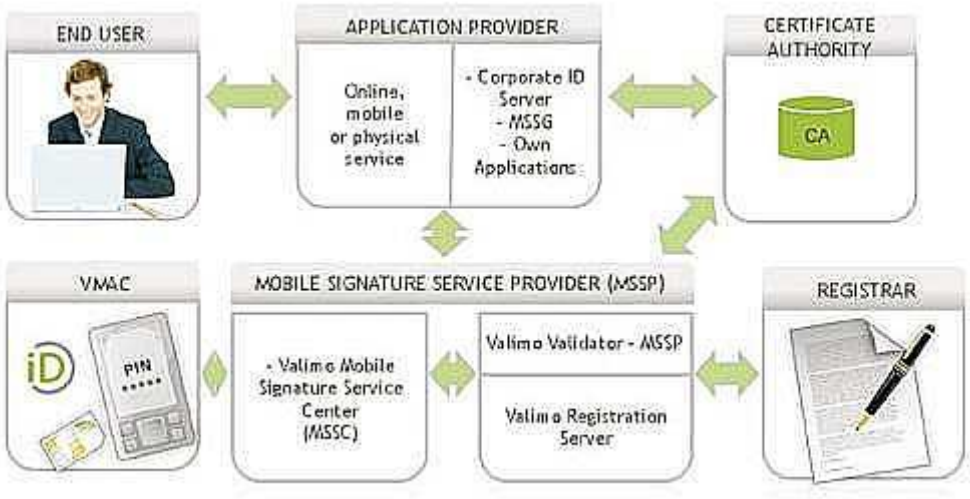

Figura 2.6: Provisión de servicio proporcionada por Valimo [16]

2.2.4.2. Trivnet

Trivnet [17] es una empresa que trabaja en el campo de los MFS (Mobile Financial Services), equipando a los operadores móviles y a los proveedores de servicios con la tecnología y el "know-how" para desarrollar este tipo de servicios y permitir que los usuarios utilicen sus teléfonos para realizar transferencias económicas, pagar facturas o realizar pedidos y compras en comercios electrónicos.

Las principales líneas de negocio de Trivnet son las siguientes:

- Dinero móvil
- Banca móvil
- Comercio electrónico

2.3. Gestión de identidad

El objetivo de la gestión de identidad es el tratamiento e intercambio de la información de identidad del usuario, más conocida como identidad digital, permitiendo la provisión de servicios fiables y seguros a los clientes a través de una arquitectura de red distribuida.

2.3.1. Identidad digital

Una identidad digital es "la representación digital de un conjunto de afirmaciones sobre un sujeto, realizadas por sí mismo o por otra entidad".

Por ello, una identidad digital es:

- Una colección de atributos o cosas que digo de mí mismo.
- Un conjunto de credenciales o cosas que los demás dicen de mí.

- Un conjunto de preferencias o cómo me presento ante los demás.

La identidad digital puede ser usada con multitud de propósitos como interactuar con otras entidades y comunidades, o dotar al sistema de mecanismos de seguridad básica (autenticación, verificación).

A continuación se detallan diferentes tipos de identidad digital [18]:

- "User-centric" o centrada en usuario: El usuario tiene control total sobre sus datos y selecciona que información proporcionar, a quien y cuando. Algunos ejemplos son las Infocards, SXIP u OpenID.

- Federada: El usuario pertenece o se encuentra asociado a una organización en la que se encuentran almacenados sus atributos y credenciales. Permite el reconocimiento del usuario en compañías que dispongan de una relación de confianza con la compañía asociada al usuario. Algunos ejemplos son SAML, OpenID o ID-WSF.

- Corporativa: Los atributos y credenciales del usuario se encuentran almacenados en una organización y dicho usuario no tiene ningún control sobre ellos. Algunos ejemplos son las cuentas de Google o la cuenta de usuario de la UC3M.

- De telecomunicación: Los identificadores son asignados a dispositivos de telecomunicaciones como un teléfono móvil o un terminal IMS (a través de su SIP uri).

La clasificación realizada anteriormente no es única, pudiendo realizarse a su vez la siguiente diferenciación [19]:

- Identidad digital 1.0:

    - Imposibilidad de elección de credenciales. El usuario se encuentra asociado a un identificador (usuario/contraseña o PKI).

    - El usuario no tiene control sobre su información de identidad.

    - Sistema no portable ni interoperable. Necesidad de crear cuentas y recordar credenciales para los diferentes servicios.

- Identidad digital 1.5:

    - Separación de autenticación de perfil de usuario.

    - Gestión de identidad para proveer al sistema de interoperabilidad, gestionar la identidad de manera descentralizada y simplificar la acción del usuario (SSO, Single Log…).

    - La identidad es compatible con numerosos servicios.

- Identidad digital 2.0:

- Identificadores re asignables que pueden ser resueltos a un conjunto de atributos (XRI/XDI).

- Numerosos mecanismos de autenticación, estableciendo diferentes niveles de autenticación y posibilitando el uso de diferentes canales.

- Posibilidad de usar diferentes credenciales de diferentes dominios.

- Centrada en el usuario. Aparece el concepto de meta-idp. Un meta-idp es una arquitectura para la gestión de identidad que permite a los usuarios utilizar diferentes identidades digitales, módulos de autenticación, protocolos de intercambio de información o protocolos de federación.

- El usuario maneja su propia información.

La versión más simple de una identidad digital consiste en un nombre de usuario y una contraseña a la que eventualmente se referirá como credencial de autenticación.

La integración de la gestión de identidad implica la integración de procesos y tecnologías en un marco único que abarca la autorización, autenticación y registro (Authentication, Authorization and Accounting).

2.3.2. Gestión de identidad federada

El objetivo principal de la gestión de identidad federada es permitir que la identidad digital (credenciales y atributos) de diferentes usuarios sea compartida e intercambiada a través de diferentes dominios de confianza de la red según las políticas establecidas. Estas políticas establecen tanto los formatos de la información a intercambiar, así como las opciones del intercambio o los requisitos de privacidad y confianza necesarios.

El uso de la gestión de identidad federada permite a su vez a los usuarios enlazar información de identidad entre cuentas sin necesidad de un almacenamiento central de información personal. Es decir, una vez que el usuario se ha autenticado con una organización o web de confianza, ya puede ser reconocido o identificado por otras compañías afiliadas que podrán solicitar contenido personal de dicho usuario sin necesidad de volver a autenticarlo.

Esta capacidad de “loguearse” una sola vez y obtener contenidos o servicios mediante el intercambio de credenciales de autenticación y perfiles de usuario entre sitios de confianza se conoce comúnmente como Single Sign-On (SSO) y será analizada más detenidamente en esta sección.

Un entorno básico de gestión de identidad federada consta de tres roles principales:

- Usuario: Destinatario final (ser humano, dispositivo o servicio) del uso de un servicio determinado.

- Proveedor de Servicios (SP): Encargado de proporcionar y gestionar a los usuarios un servicio determinado.

- Proveedor de Identidad (IdP): Son autoridades de autenticación encargadas de autenticar a los usuarios que desean acceder a los SPs. Pueden delegar el mecanismo de autenticación propiamente dicho a otros componentes de red tales como RADIUS o LDAP. El proceso de autenticación raramente revela la identidad real del usuario para preservar su privacidad, por lo que es común el uso de alias.

Para el correcto funcionamiento del mecanismo de gestión de identidad federada deberá existir una relación de confianza entre el IdP y el SP. El conjunto de entidades que comparten relaciones de confianza se denominará comúnmente círculo de confianza.

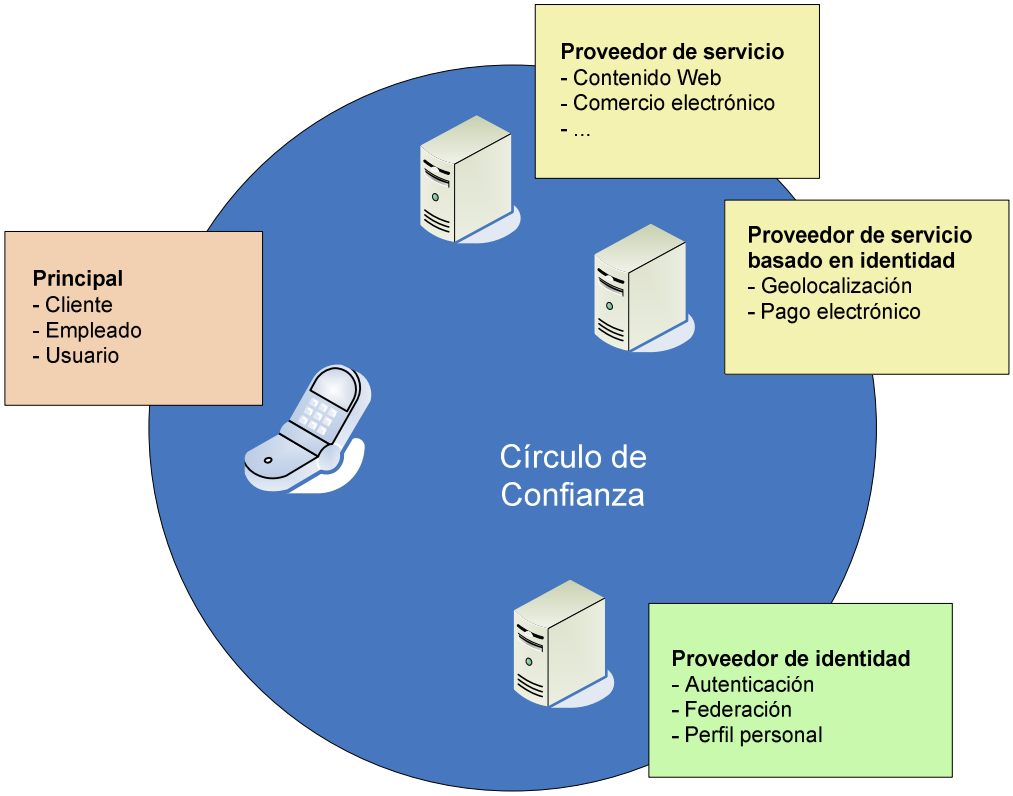

Figura 2.7: Círculo de confianza en los sistemas de gestión de identidad

El mecanismo de gestión de Identidad federada tan sólo estandariza las peticiones y respuestas de autenticación, dejando a elección del implementador el resto de parámetros, tales como método de autenticación, políticas de generación de pseudónimos o elección de identidades para un SP determinado, lo que dota al sistema de una gran libertad y flexibilidad de uso.

2.3.3. Gestión de identidad en redes de próxima generación

En las redes de telecomunicaciones tradicionales, la gestión de los usuarios se hace de forma separada. Es decir, si un proveedor de servicio ofrece servicio de línea telefónica, acceso a Internet y televisión por cable, gestiona individualmente los perfiles de usuario, almacenando la información de los usuarios en bases de datos independientes.

En las redes de próxima generación, el objetivo es unificar el soporte a los diferentes servicios provistos por los operadores, para mejorar la personalización y aumentar la experiencia del usuario. Por ello, la gestión de la identidad se convierte en un factor clave de desarrollo [6][20].

La gestión de identidad conlleva la conformación, tratamiento e intercambio de la información de identidad de los usuarios para proporcionarles un acceso rápido, seguro y fiable a la red.

La información de identidad en una red de próxima generación puede incluir una combinación de nombres, identificadores de usuario, identificadores de terminal, direcciones, credenciales o perfiles de usuario, etc. La identidad digital del usuario dependerá a su vez del ámbito en el que se vaya a usar, por lo que varios perfiles podrán definirse para el usuario.

El perfil personal recoge las preferencias del usuario, así como información acerca de suscripción a servicios, atributos de los terminales del usuario, intereses, recursos disponibles o grupos a los que pertenece. La información almacenada y compartida en los perfiles personales es variable en función de las aplicaciones o entornos en los que se encuentre el usuario.

Una buena gestión de identidad permite al usuario acceder de forma ubicua a aplicaciones o servicios a través de diferentes proveedores de servicios sin tener que realizar varias operaciones de login, ni recordar un nombre de usuario y contraseña para cada uno de ellos. Además hace posible la reducción de costes en la provisión de servicio y centraliza la gestión de la seguridad de la identidad del usuario.

2.3.4. Single Sign-On (SSO)

Single Sign-On (SSO de ahora en adelante), es un procedimiento de autenticación que permite a un usuario acceder a varios sistemas o servicios con una sola instancia de identificación. El usuario se identifica por primera vez al acceder a una determinada aplicación. Posteriormente, el usuario permanecerá autenticado a lo largo de la sesión iniciada y podrá acceder a otras aplicaciones sin necesidad de identificarse de nuevo.

Existen cinco tipos principales de SSO:

- E-SSO (Enterprise SSO): Realiza una autenticación primaria, interceptando los requisitos de login de las aplicaciones para completarlos con el nombre de usuario y la contraseña.

- Web-SSO: Trabaja con aplicaciones y recursos accedidos a través de la Web. Los accesos son interceptados mediante un Proxy que redirige a los usuarios no autenticados a un servidor de autenticación. Permite el uso de cookies para mantener el estado de autenticación asociado a un usuario.

- Kerberos: Los usuarios se autentican en el servidor Kerberos y reciben un “ticket” que posteriormente presentan las aplicaciones cliente para obtener acceso.

- Identidad federada: Trabaja, al igual que Web-SSO, con aplicaciones Web. Hace uso de protocolos basados en estándares para evitar la autenticación redundante.

- Asociado a OpenID: Proceso de SSO distribuido y descentralizado en el que la identidad digital se asocia a una URL que será verificada por un servidor de identidad.

Existen numerosas implementaciones del procedimiento SSO. Con el fin de explicar de manera más detallada su funcionamiento a continuación se proporciona una descripción de la especificación SSO realizada por el producto OpenSSO (Sun Microsystems-Oracle). A su vez el funcionamiento de OpenSSO se detallará en la sección 2.3.5.

Según la implementación realizada por OpenSSO, el procedimiento SSO puede establecerse entre aplicaciones pertenecientes al mismo dominio (SSO intradominio), o entre aplicaciones alojadas en diferentes dominios (SSO interdominio). A continuación se describen ambas especificaciones:

SSO intradominio:

Se define como la infraestructura necesaria para el establecimiento de sesiones SSO entre aplicaciones web pertenecientes al mismo dominio, delegando las peticiones HTTP que se envían entre ellas a un agente SSO propio.

El agente SSO de una aplicación se encargará de comprobar la existencia de una sesión previa de usuario mediante el uso de cookies, y en caso contrario de presentarle un formulario de login. Una vez autenticado, el usuario tendrá acceso a todas las aplicaciones del dominio de confianza mediante la sesión SSO iniciada.

La existencia de un agente en el sistema que actúa como intermediario viene motivada por la necesidad de proporcionar diferente tipo y cantidad de información del perfil del usuario en función de la aplicación que acceda al mismo. En numerosas ocasiones se hace uso de directorios LDAP para agilizar el procesamiento y el tiempo de acceso a las aplicaciones.

El proceso para acceder a una aplicación web intradominio que soporta SSO sería:

1. El cliente envía una petición HTTP a través del navegador incluyendo una cookie en caso de que el usuario ya se encuentre autenticado.

2. El agente SSO examina la cookie:

    2.1. Si no existe redirige al usuario al formulario de login.

    2.2. Si existe extrae el token asociado para validar su autenticidad.

3. El agente SSO analiza el token:

    3.1. Si no es válido será necesaria la autenticación y se solicitará autenticarse al usuario.

3.2. Si es válido se crea un "*session listener*" (entidad que se mantiene a la escucha de los eventos contenidos en una sesión y que envía notificaciones sobre los mismos al "policy agent" asociado) para mantener al servidor OpenSSO actualizado frente a los cambios que se produzcan en la sesión.

4. El agente SSO pregunta al servidor de autorización acerca de los permisos que el usuario posee en su perfil para el recurso solicitado:

4.1. Si no tiene permiso para acceder a dicha información se le deniega el acceso.

4.2. Si tiene permiso se le redirige a la página solicitada.

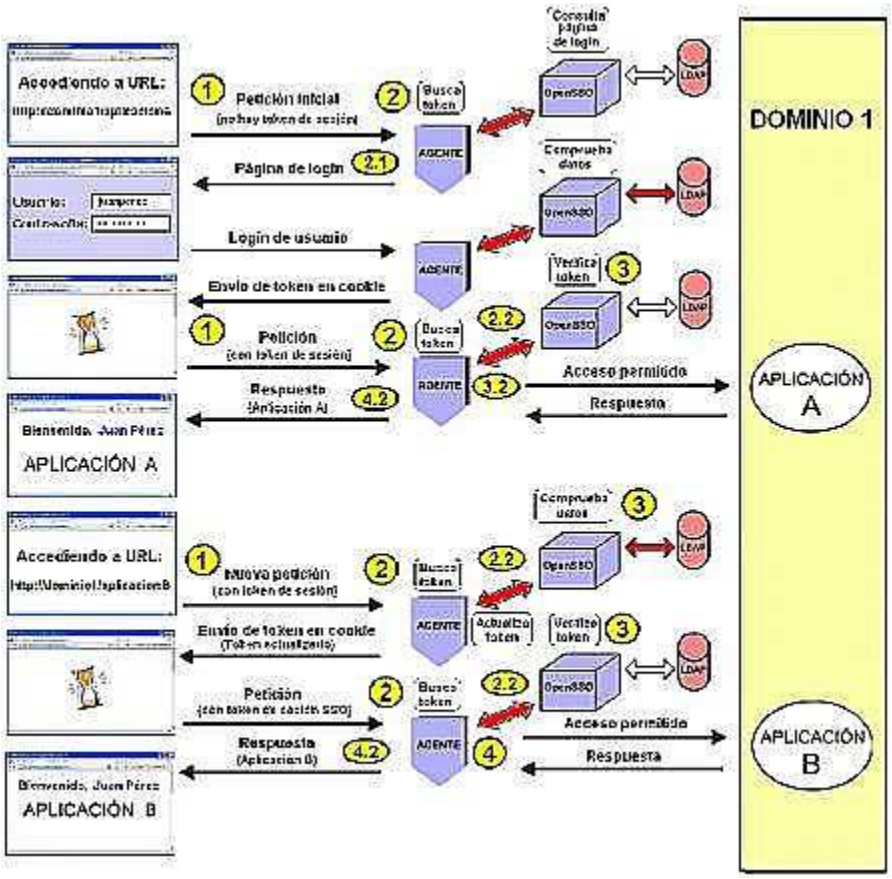

Figura 2.8: Diagrama de autenticación SSO intradominio. Obtenido de [21].

SSO interdominio:

En el caso de que las aplicaciones web a las que queremos dotar de soporte SSO se encuentren en dominios diferentes, es decir, dispersas por Internet, la gestión es más complicada y se requiere que las aplicaciones estén federadas. Este tipo de gestión de identidad se conoce como CD-SSO (Cross Domain SSO).

La información de identidad intercambiada entre el proveedor de identidad y los proveedores de servicios debe atenerse a un protocolo dado, como por ejemplo ID-FF, Shibboleth, SAML, etc.…

El proceso para acceder a una aplicación web interdominio que soporta SSO sería:

1. El agente SSO asociado a la aplicación solicitada solicita y examina el token de sesión.

    1.1. Si existe token, el agente SSO valida la sesión SSO.

    1.2. Si no existe token, se redirige al usuario al formulario de autenticación o en caso de no disponer de los permisos pertinentes se muestra un mensaje de error.

2. El agente SSO redirige la petición al servidor CD-SSO.

3. El servidor SSO extrae la información de la sesión a partir del token y la envía al agente SSO.

4. El agente SSO utiliza la información de la sesión para validarla estableciendo un nuevo token para que la sesión sea reconocida por los agentes del nuevo dominio.

5. Se valida la autorización y las operaciones solicitadas:

    5.1. Si se deniega el acceso se redirige al usuario a una página de error.

    5.2. Si se le proporciona el acceso se le redirige a la página del recurso solicitado.

2.3.5. OpenSSO

OpenSSO [22] es una plataforma de código libre nacida en Julio de 2005 para la gestión de identidad. Como producto, OpenSSO es considerado uno de los mayores esfuerzos realizados en materia de gestión de identidad, estableciendo un meta-IDP en el que pueden integrarse multitud de protocolos de intercambio de información, protocolos de federación o módulos de autorización y autenticación.

La plataforma, perteneciente al proyecto Open Web SSO bajo licencia CDDL (Common Development and Distribution License), se basa en los gestores de acceso y federación de Sun Microsystems para proporcionar diferentes servicios [18]:

- SSO (Single Sign-On) y control de acceso:
    - Intercepta el acceso a un recurso.
    - Autentica al usuario y en caso de resultado positivo, genera un token. Existen numerosos mecanismos de autenticación disponibles (LDAP, RADIUS, RSA, SecureID, etc.) así como pueden definirse otros personalizados.

- Evalúa las políticas asociadas con el recurso solicitado.

- Si el usuario está autorizado le proporciona acceso al recurso y a la información asociada al mismo.

- Repite el proceso continuamente.

- SSO (Single Sign-On) federado:

  - Servicio integrado en la iniciativa sin necesidad de software adicional.

  - El proveedor de servicio envía solicitud de autenticación al cliente.

  - El cliente se redirige al proveedor de identidad que tiene asociado.

  - El usuario es autenticado.

  - El proveedor de identidad genera un aserto que el usuario entrega al proveedor de servicio.

  - Permite la unión de respectivas cuentas en el proveedor de servicios y en el proveedor de identidad preservando la privacidad del usuario mediante identificadores únicos y opacos.

- Seguridad en Servicios Web:

  - Identificar al usuario final y al servicio web participante.

  - Preservar la identidad extremo a extremo del usuario, tanto dentro de su dominio y fuera de él basándose en los estándares disponibles.

  - Uso de contenedores para la generación y validación de tokens de seguridad (GlassFish, WebSphere, etc.…).

- Servicios asociados a la gestión de identidad:

  - Autenticación.

  - Autorización.

  - Auditoría.

  - Provisión de atributos.

- Servicios de interoperabilidad.

  - Orientado a los desarrolladores para simplificar la seguridad.

- Reutilización de servicios.

- Soportado en numerosos IDE de desarrollo (Netbeans, Eclipse, Visual Studio, etc.…).

2.3.6. OpenID

OpenID es un estándar de facto que permite gestionar la identidad de manera descentralizada nacido en 2005. Existen numerosas implementaciones de dicho estándar, siendo algunas de ellas de código abierto.

Un sistema de gestión de identidad mediante OpenID constará de los siguientes elementos:

- Proveedor de identidad (IdP): Se encargará de gestionar los perfiles de usuario e identificar a los mismos para poder acceder a un determinado servicio.

- Proveedor de servicio (SP): Se encargará de solicitar información de identidad al IdP para identificar a los usuarios que deseen acceder a un servicio determinado.

- Relying party (RP). Entidad que se encarga de comunicar los determinados SPs con el IdP. Para ello, se encarga de crear las peticiones de identificación o de intercambio de atributos y posteriormente recibir las respuestas provenientes del IdP para presentárselas al SP que hizo la solicitud.

Para su autenticación mediante OpenID, los usuarios tan sólo deben poseer una cuenta en un proveedor de identidad de OpenID, a la que accederán mediante un identificador, que normalmente será una URL o un XRI.

Además, los sitios web a los que deseen acceder los usuarios mediante OpenID deberán soportar este mecanismo de gestión de identidad, ya que una vez introducido el identificador de OpenID del usuario, éste será verificado por un servidor que soporte el protocolo.

OpenID permite al sistema de gestión de identidad demostrar quién es el propietario de cierto identificador. La seguridad del sistema viene dada por la relación de confianza establecida entre el usuario y el proveedor de identidad OpenID y, dependiendo del nivel de confianza establecido con dicho proveedor, el usuario podrá ser habilitado para realizar ciertas operaciones o no.

A su vez, a través de la extensión PAPE (Provider Authentication Policy Extension) [23], OpenID permite autenticar al IdP y al SP. PAPE posibilita a un determinado RP solicitar a un proveedor de identidad OpenID el uso de unas determinadas políticas en el proceso de autenticación o que le comunique los niveles de autenticación usados.

En el caso de requerirse el envío de información adicional a la comentada anteriormente OpenID permite el uso del mecanismo de Intercambio de Atributos (Attribute

Exchange) [25]. Dicho mecanismo para el intercambio de información permite almacenar y recuperar información de identidad entre elementos finales.

Por otro lado, el carácter descentralizado de OpenID hace posible que cualquiera pueda actuar como proveedor de identidad, dejando al usuario la posibilidad de elegir aquél que le proporcione una mayor confianza o incluso varios de ellos sin la necesidad de utilizar varios identificadores.

Algunos de los principales servicios derivados del uso de OpenID son:

- Autorización.
- Autenticación [25].
- Gestión de perfil de usuario (intercambio de atributos) [26].
- Descubrimiento de servicios (Yadis) [26].
- Registro simple.
- Modelo de confianza y seguridad (extensión PAPE) [23].

Otro valor añadido al uso de OpenID es su versatilidad, ya que es un caso especial dentro de los estándares de gestión de identidad. En cierto modo es un sistema de gestión de identidad centrada en el usuario, lo que supone que la información a proporcionar a terceros será totalmente gestionada por dichos usuarios. El carácter abierto de OpenID hace que la información almacenada por un usuario en cierto proveedor no tenga porque ser la misma que la almacenada en otro.

Por otro lado, OpenID se encuentra en periodo de expansión, en busca de una posición dominante en el marco de la gestión federada en Internet, ya que numerosos sitios web de contrastado prestigio han adoptado ya dicho mecanismo de identificación. En España, Movistar es el operador pionero en el desarrollo de la gestión de identidad mediante OpenID.

En la actualidad existen gran cantidad de proveedores de identidad tales como AOL, myOpenID, Google, Flickr, Yahoo, etc.…además de varias implementaciones de OpenID de código abierto, tales como OpenID4Java, JOpenID o WSO2 OpenID, que permiten la creación de IdP's personalizados.

Funcionamiento del sistema:

Para que su identificación mediante OpenID, el usuario debe registrarse en un proveedor de identidad OpenID previamente al proceso de autenticación. Para ello, deberá abrir una cuenta en un proveedor de identidad de su confianza y modificar su perfil de usuario con los atributos que desee sean accesibles a los proveedores de servicio.

El usuario accederá al formulario de identificación de la página web a la que desea acceder. En el caso de identificación mediante OpenID, en lugar del formulario tradicional que solicita nombre de usuario y contraseña, tan sólo se requerirá un campo, que o bien será una

XRI del tipo `identificador.proveedor_openid` o una URL, encargándose la propia página web de transformarla en un identificador del tipo: `http://identificador.proveedor_openid/`. En el caso de la versión 2.0 de la autenticación con OpenID, el usuario puede introducir directamente un identificador personalizado, como por ejemplo: `web_destino.identificador`, que la página web también se encargará de adaptar al formato requerido.

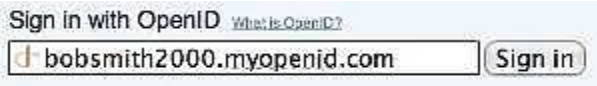

Figura 2.9 Identificador OpenID asociado a un usuario. Obtenida de [28].

El proveedor de servicio (la página web a la que el usuario desea acceder), realizará un descubrimiento de la localización del proveedor de identidad asociado al usuario a partir del identificador introducido mediante el descubrimiento por URL o descubrimiento Yadis [26].

Opcionalmente, el proveedor de servicio podrá establecer una asociación con el proveedor de identidad estableciendo un secreto compartido entre ellos. Dicha asociación permitirá agilizar posteriores operaciones entre el RP y el proveedor de OpenID. Para el establecimiento de dicha asociación se hace uso del siguiente modo:

- associate: Establece un secreto compartido entre el usuario y el proveedor de identidad.

Una vez se tiene una asociación de seguridad (secreto compartido) el SP puede comprobar la "claimed id" (identidad asociada a un usuario) con los métodos:

- checkid_immediate: Método usado en caso de que se desee que no haya interacción entre el usuario y el proveedor de identidad. El RP pregunta directamente al IdP si un usuario final posee un determinado "claimed id" y obtiene como respuesta inmediata "yes" o "can't say".

- checkid_setup: Método usado en caso de que se desee que haya interacción entre el usuario y el proveedor de identidad. Pregunta al IdP si un usuario final posee un determinado "claimed id" y espera a la respuesta. El cliente se comunica a través del User Agent con el IdP por un corto período de tiempo en el que obtendrá una respuesta de "yes" o "cancel".

Posteriormente, si se necesita intercambiar atributos, puede hacerse uso del módulo Attribute Exchange de OpenID. Su uso consiste en el envío de un "fetch message", que permite obtener atributos pertenecientes a la identidad del usuario del proveedor OpenID. A continuación se muestran los principales campos de un "fetch message":

- openid.ax.mode: Parámetro establecido al valor "fetch request", que indica que la petición se encuentra en modo de petición de intercambio de atributos. Este campo es obligatorio.

- openid.ax.type.<alias>: Parámetro que indica el tipo de identificador (mediante una URI) de un atributo determinado. Este campo es opcional.
- openid.ax.required: Parámetro que describe una lista de atributos asociados a un tipo determinado por el campo anteriormente descrito y que serán remitidos en la petición. Este campo es opcional.

El proveedor de servicio redirige al usuario al proveedor de identidad donde, en caso de no tener activa una sesión con dicho proveedor, deberá autenticarse mediante usuario y contraseña o cualquier otro mecanismo de autenticación (puede variar en algunas implementaciones). Una vez autenticado contra el proveedor de identidad, se solicitará permiso al usuario (autorización) para realizar las operaciones solicitadas por el proveedor de servicio.

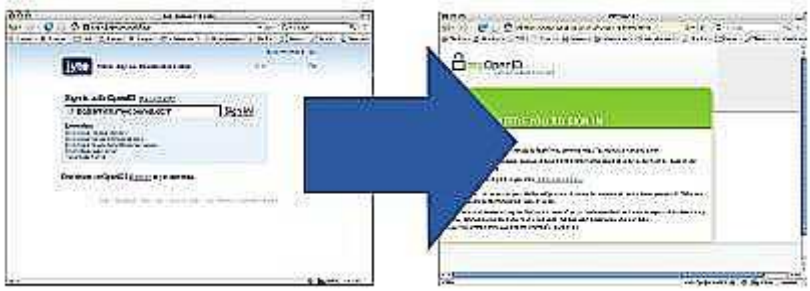

Figura 2.10 Redirección de usuario a su proveedor de identidad. Obtenida de [28].

Una vez recibida la autorización, el usuario es redirigido nuevamente a la página web. Posteriormente, en caso de haber solicitado el intercambio de algún atributo del usuario adicionalmente a la autenticación, el proveedor de identidad se los enviará en una respuesta, siempre y cuando el usuario lo haya autorizado (se le preguntará nuevamente para ello).

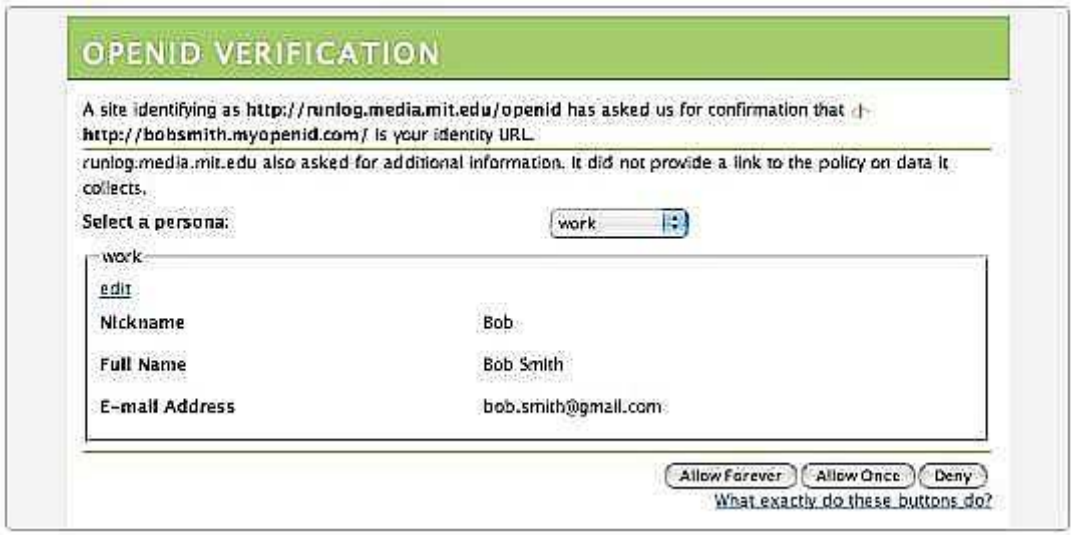

Figura 2.11: Perfil de usuario en el proveedor de identidad. Obtenida de [28].

Seguidamente, el proveedor de servicio verifica la respuesta obtenida para comprobar si las credenciales recibidas provienen realmente del proveedor de identidad correspondiente.

En caso de poseer el secreto compartido generado en la asociación inicial entre proveedores de servicio y de identidad, el primero comprobará las credenciales existentes de dicha asociación con las obtenidas en la respuesta de autenticación de manera directa. En caso contrario, deberá hacerse uso del siguiente modo para realizar la verificación solicitando información al proveedor de identidad OpenID:

- check_authentication: Pregunta al IdP si un mensaje es válido. Este mecanismo es válido específicamente para respuestas sin asociaciones establecidas.

Finalmente, la página web redirige al usuario a la página de retorno establecida en caso de autenticación satisfactoria, o se notificará al proveedor de servicio a través del RP en caso de no completarse exitosamente la autenticación.

Una vez autenticado a través del proveedor de identidad, el mismo enviará confirmaciones al proveedor de servicio automáticamente sin necesidad de autenticarse de nuevo (Single Sign-On).

El esquema global del mecanismo de autenticación mediante OpenID se muestra a continuación:

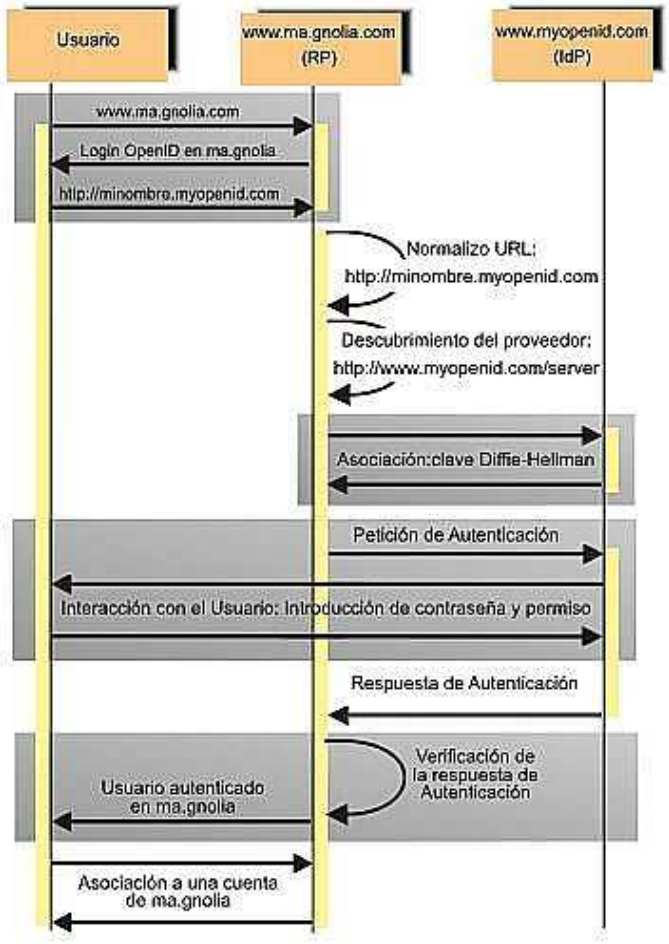

Figura 2.12: Diagrama de flujo de identificación mediante OpenID. Obtenida de [28].

Beneficios derivados del uso de OpenID:

- Simplicidad del proceso de autenticación en las páginas web, permitiendo la identificación mediante el simple uso de un identificador como una XRI. Además, mediante el uso de SSO, elimina la necesidad de actualizar múltiples perfiles de usuario, centralizando la información en el proveedor de identidad.

- Crecimiento del estándar en los últimos años y la consecuente aparición de numerosas implementaciones, entre las que se encuentran algunas especificaciones de código abierto.

- Versatilidad del estándar de gestión de identidad: Permite el uso descentralizado de la identidad sin control por parte de los proveedores de servicio (gestión de identidad centrada en usuario). El usuario decide la información que quiere compartir con cada proveedor de servicio.

  Por otro lado, permite la gestión de identidad federada en la que distintas aplicaciones establecen una relación de confianza con el IdP para intercambiar información de identidad del usuario.

  Vulnerabilidades derivadas del uso de OpenID:

- Ineficiencia para escenarios que requieren de mecanismos de seguridad estrictos.

2.4. Tecnologías de seguridad

A lo largo del texto vamos a tratar el uso de criptografía para cifrar los mensajes de autenticación, por lo que haremos una breve introducción del sistema RSA que será el utilizado en nuestro caso.

A su vez, al ser el intercambio de mensajes realizado en formato XML, definiremos el formato de un documento XML firmado digitalmente [28].

2.4.1. Algoritmo de cifrado RSA

Algoritmo de cifrado asimétrico de clave pública creado por Rivest, Shamir y Adelman [29] en 1978, con la característica de ser reversible, es decir, permite cifrar con la clave pública y descifrar con la clave privada y viceversa.

Proporciona varios servicios:

- Firma digital.

- Cifrado.

- Verificación de credenciales.

- Intercambio de claves.

RSA normalmente es usado tanto para obtener confidencialidad (cifrando con la clave pública del destinatario), como para firmar (cifrando con la clave pública del emisor).

A modo práctico, el funcionamiento básico del cifrado RSA es el siguiente:

Supongamos que un usuario (llamado Bob) quiere enviar un mensaje M a una persona (llamémosla Alice). Simplemente necesita obtener la clave pública de Alice (n,e) y luego calcular el mensaje cifrado c:

$c = Me \bmod(n)$

Luego, Bob envía el mensaje c a Alice, quien es capaz de descifrarlo con su clave privada (p,q,d):

$M = Me*d \bmod(n) = cd \bmod(n)$

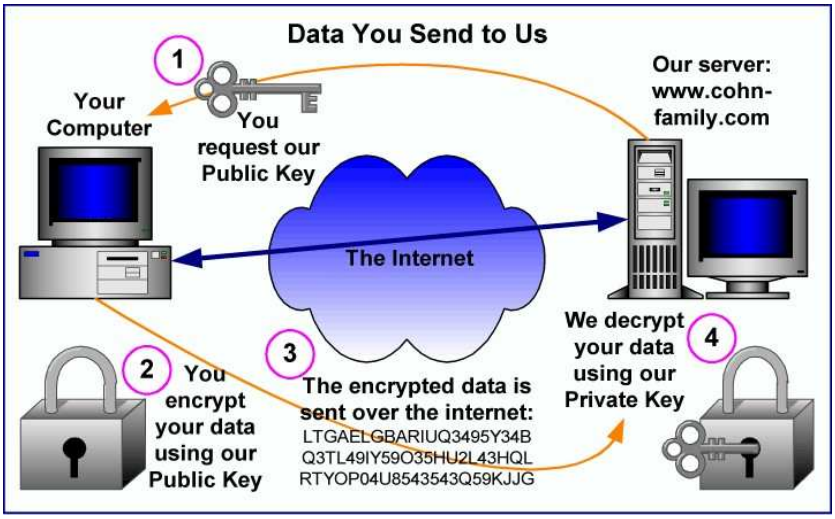

Figura 2.13: Funcionamiento del cifrado RSA. Obtenida de [30].

2.4.2. Firma digital de documentos XML

Como se ha especificado anteriormente, la necesidad de garantizar la confidencialidad e integridad de los datos que viajan a través de Internet, es requisito fundamental. Dichos datos presentan diversidad de formatos, requisitos y estructuras, por lo que se necesita una gestión estandarizada de los mismos.

Como respuesta a los problemas presentados, nacen estándares como XML Encryption (cifrado) y XML Signature (firma digital), orientados a manejar situaciones en las que un documento puede requerir diferenciación en el trato de cada una de sus partes (tomando por ejemplo documentos en los que parte del contenido esté orientado a todo tipo de remitentes y parte del mismo esté restringido a un grupo de usuarios). El uso de cifrado es clave en dicho contexto, ya que es lo que va a confirmar la identidad del texto. Las firmas digitales, por su parte, permitirán la verificación del remitente.

- **XML Encryption:** Es un lenguaje orientado a proveer confidencialidad a documentos XML, cifrando el contenido del mensaje a transmitir.

  A modo de ejemplo mostramos el proceso de cifrado del siguiente documento XML, que representa el pago de un usuario con una tarjeta de crédito:

```
<?xml version='1.0'?>
  <Metodopago xmlns='http://ejemplo.org/pago'>
    <Nombre>Cliente Ficticio</Nombre>
    <TarjetaCredito Limite='5.000' Moneda='EU'>
      <Numero>0000 0000 0000</Numero>
      <Issuer>Ejemplo de Banco</Issuer>
      <Caducidad>10/05</Caducidad>
    </TajetaCredito>
  </Metodopago>
```

  Como hemos podido observar, en el documento anterior, la información del número de la tarjeta de crédito no se encuentra protegido, por lo que el uso de XML Encryption es necesario:

```
<?xml version='1.0'?>
   <Metodopago xmlns="http://ejemplo.org/pago">
    <Nombre>Cliente Ficticio</Nombre>
    <DatosEncriptados xmlns="http://www.w3.org/2001/04/xmlenc#"
      Tipo="http://www.w3.org/2001/04/xmlenc#Element">
      <DatosClave>
        <DatosClave>A23B45C56</DatosClave>
      </DatosClave>
    </DatosEncriptados>
  </Metodopago>
```

  Una vez cifrado, el elemento tarjeta de crédito queda oculto, siendo imposible determinar si el usuario ha empleado un método de pago u otro. El elemento `DatosClave` contiene el número cifrado del elemento `TarjetaCrédito`.

- XML Signature: Es un protocolo orientado a garantizar la integridad en el intercambio de documentos XML. A su vez, proporciona el servicio de autenticación de firma, tanto si ésta se encuentra dentro del documento XML que la incluye o en otro lugar. Además, existe la posibilidad de firmar sólo las partes especificadas del documento, lo que resulta interesante en aplicaciones donde la persona que crea el documento no es la misma que lo firma.

  Su funcionamiento se basa en la asociación de claves con datos de consulta, representado un sistema que ofrece la autenticidad de los datos mediante firma digital, que confirma la identidad del emisor, la autenticidad del mensaje, su integridad, e incluso la garantía de no repudio.

  A continuación se muestra la estructura básica de un documento firmado mediante XML Signature:

```
<Signature Id="EjemploXMLSignature"
 xmlns="http://www.w3.org/2000/09/xmldsig#">
   <SignedInfo>
     <CanonicalizationMethod
```

```
      Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
     <SignatureMethod
      Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
     <Reference
        URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
        <DigestMethod
          Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
      </Reference>
   </SignedInfo>
   <SignatureValue>MC0CFFrVLtRlk=...</SignatureValue>
   <KeyInfo>
      <KeyValue>
         <DSAKeyValue>
           <p>...</p><q>...</q><g>...</g><y>...</y>
         </DSAKeyValue>
      </KeyValue>
   </KeyInfo>
</Signature>
```

El elemento `Signature` encapsula la firma digital y contiene a los sub-elementos `SignedInfo`, `SignatureValue` y `KeyInfo`. El primero de ellos es el que contiene la información necesaria para la creación y la validación de la firma. A su vez, éste elemento contiene dos sub-elementos: `CanonicalizationMethod` y `SignatureMethod`, que indican el algoritmo aplicado para transformar SignedInfo antes de realizar la firma digital y el algoritmo utilizado para calcular el valor de dicha firma, respectivamente. A su vez, en `SignedInfo` también se incluyen las referencias a los objetos firmados (`Reference`), que a su vez incluirá `DigestMethod` y `DigestValue`. La validación de una firma requiere dos procesos de validación diferenciados: el de la firma propiamente dicha, y el de los resultados de las referencias.

Cada elemento `Reference`, incluye una referencia al objeto que se firmará. Al mismo tiempo incluye el resultado de `DigestValue` que es el valor resultante.

`SignatureValue` contendrá el resultado obtenido tras la generación de la firma, una vez ejecutados los algoritmos mencionados anteriormente sobre el elemento `SignedInfo`. El resultado de la firma se encontrará codificado y contendrá un atributo único que servirá para identificar la firma en procesos de validación asociados.

`KeyInfo` es un elemento opcional que indica la clave que ha de utilizarse para validar la firma. El elemento `KeyValue`, a su vez, especifica la clave para validar la firma digital.

La entidad receptora de este documento, deberá ser capaz de verificar la firma del certificado digital mediante la información obtenida al acceder a los elementos comentados con anterioridad.

2.5. Redes de próxima generación (NGN)

El proyecto G11 de la ITU-T define una NGN (Next Generation Network) como "la provisión de un conjunto de servicios por parte de un conjunto de proveedores de servicio sobre una variedad de tecnologías de red provenientes de diferentes sectores de la industria".

Bajo el concepto de NGN, múltiples servicios, como telefonía, Internet, video bajo demanda (VoD) y demás aplicaciones multimedia, junto con nuevos servicios emergentes, convergerán en una misma red. Las redes de circuito de telecomunicaciones evolucionarán a las redes de paquetes.

Algunos de estos servicios dependerán de proveedores de servicio externos. Los usuarios podrán acceder a la red y a sus servicios desde diferentes localizaciones usando gran variedad de redes de acceso, tecnologías y terminales.

El concepto de NGN es acuñado generalmente en torno a varios términos fundamentales como son: movilidad, servicio personalizado y servicio basado en localización.

Movilidad:

El término se refiere a la capacidad del usuario para cambiar su terminal o punto de acceso a la red manteniendo una consistencia en el servicio de la misma. Dentro de la movilidad podemos diferenciar:

- Movilidad personal: Capacidad para cambiar localización y/o técnica de acceso a la red (WLAN, Bluetooth, UMTS,...)
- Movilidad de terminal: Capacidad para cambiar localización o técnica de acceso sin necesidad de cambiar de terminal.
- Nomadismo: Capacidad para cambiar el punto de acceso a la red al cambiar de localización, pero sin posibilidad de mantener la sesión de servicio.
- Roaming: Capacidad para acceder a la red subscrita a partir de otra red

Servicio personalizado:

Es la capacidad de proveer a cada usuario de los servicios que más se ajusten a sus necesidades y a sus condiciones en un momento determinado. La personalización del servicio es un término muy subjetivo, ya que puede interpretarse de maneras muy diferentes dependiendo del usuario. Para la provisión de la personalización se requiere el uso de Identificación y de Perfiles de Usuario.

Servicio basado en localización:

Capacidad para identificar la localización geográfica del dispositivo móvil y así proveer al mismo con la información y los servicios disponibles en base a su posición. Por ejemplo, si se encontrase un usuario en una determinada posición y solicitase a la red la situación de un

hospital, tan sólo debería recibir información acerca de los hospitales más cercanos a la misma.

2.5.1. IMS (IP Multimedia Subsystem)

Se define como una arquitectura de red para el desarrollo de servicios multimedia sobre IP. Originalmente fue diseñado por el 3GPP [31] con el fin de promover la evolución de las redes móviles más allá de GSM y desarrollando servicios de Internet sobre GPRS. Posteriormente esta visión fue actualizada por 3GPP, 3GPP2 y TISPAN, requiriendo soporte para otras redes como WLAN, CDMA2000 o línea fija.

IMS forma parte del núcleo de la arquitectura de las redes de próxima generación. Estas redes son capaces de proporcionar servicios multimedia fijos y móviles.

Los usuarios pueden conectarse a estas redes de múltiples maneras, la mayoría de las cuales usan el protocolo IP. Algunas de las posibilidades son: terminales IMS (teléfonos móviles, PDAs, ordenadores), puntos de acceso fijos (DSL, Ethernet), puntos de acceso móviles (W-CDMA, GSM, GPRS) o puntos de acceso inalámbricos (WLAN, WiMAX). A su vez los usuarios pueden conectarse a la red a través de dispositivos VoIP no compatibles con IMS o la red telefónica analógica a través de gateways.

Una sesión multimedia entre un usuario conectado a la red IMS y un usuario conectado a Internet se efectúa usando los mismos protocolos que una entre dos usuarios de Internet. Los protocolos usados por IMS están aprobados por IETF y entre ellos uno de los más importantes es SIP (Session Initiation Protocol) [30], ya que proporciona una manera fácil y flexible de controlar aplicaciones multimedia sobre una red IP, haciendo que la convergencia entre las redes de telecomunicaciones e Internet sea posible.

El objetivo de estas redes no es la simple prestación de servicios, sino la funcionalidad para todo tipo de servicios, actuales y futuros, que puedan prestarse en Internet. Con ello permiten que los operadores de ISP puedan controlar y facturar cada uno de los servicios y que los usuarios tengan acceso a los mismos independientemente de si su localización es fija o se encuentran en movimiento. Para ello se basará en el modelo de Red Convergente:

- Convergencia de servicios: Proporciona nuevos medios funcionales para los usuarios.
- Convergencia de medios: Permite acceder a diferentes tipos de multimedia (voz, datos, video, texto, imágenes).
- Convergencia de redes: Permite acceder a diferentes servicios a través de diferentes tecnologías de red.
- Convergencia de dispositivos: Proporciona un nivel de estandarización y descripción que permite usar los mismos servicios con diferentes dispositivos dando lugar a soluciones integrales de menor coste.

Estructura de IMS:

A nivel estructural, IMS se compone de tres capas, como vemos en la Figura 2.14, que se ensamblan para proporcionar servicios multimedia de calidad:

- Capa Servidor de Aplicaciones: En ella residen las aplicaciones IMS.
- Capa de Control: Funciones de control SIP/H.323 para el acceso a las aplicaciones o la administración de servicios y usuarios.
- Capa de Transporte y acceso: Provee conectividad a los diferentes dispositivos en la propia red y con otras redes.

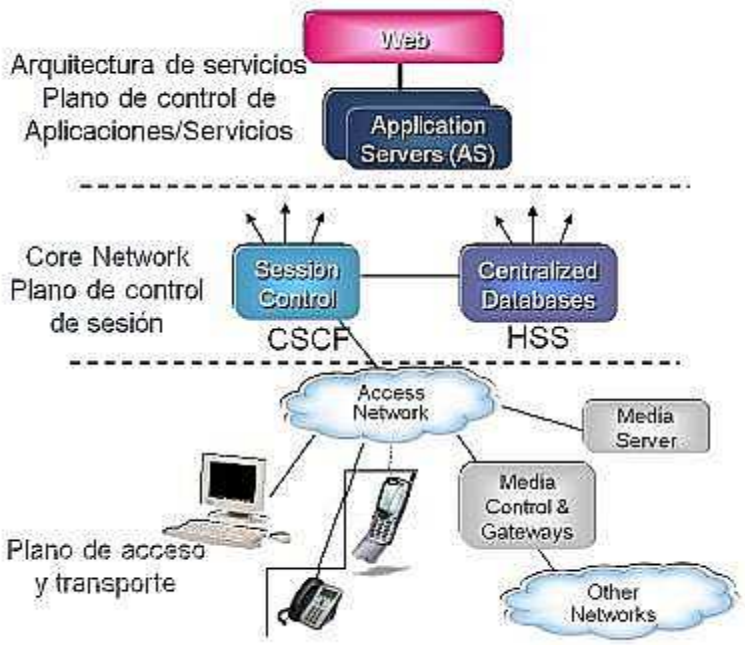

Figura 2.14: Diagrama estructural de IMS. Obtenido de [33].

Elementos funcionales de IMS:

A continuación, se describen los principales elementos funcionales de IMS, que podemos observar en Figura 2.14:

- Application Server (AS): Ejecutan servicios e intercambian señalización mediante el protocolo SIP.
- Call Session Control Function (CSCF): Conjunto de elementos que actúan como proxies SIP, encaminando las llamadas, mensajes y señalización que viaja por la red.

- P-CSCF (Proxy CSCF): Sirve como primer punto de contacto. Autentica al usuario y puede aplicar políticas.
- I-CSCF (Interrogating CSCF): Actúa como filtro de entrada a la red, aislándola del exterior.
- S-CSCF (Serving CSCF): Procesa las peticiones de registro de los usuarios e invoca a los servicios alojados en los AS.

- Home Subsriber Server (HSS): Almecena la información de usuario referente a suscripciones y así habilitar los procesos de autorización y autenticación.

Identificación de usuarios en IMS:

Existen diferentes representaciones de la identidad asociada a un usuario dentro de una red IMS, entre las que podemos destacar:

- IMPI (IP Multimedia Private Identity): Identidad única, global y permanente asociada por el operador de la red.
  - Usada en el proceso de Registro, Autorización, Administración y Accounting.
  - Un usuario puede tener una o varias IMPI asociadas.
- IMPU (IP Multimedia Public Identity): Identificador utilizado por los usuarios para solicitar comunicaciones con otros usuarios.
  - Pueden asociarse varios IMPU a un mismo IMPI.
  - El IMPU puede usarse en varios dispositivos y compartirse su uso por varios de ellos al mismo tiempo.

Despliegue de aplicaciones convergentes en IMS:

IMS, como modelo de red convergente, permite que diferentes tipos de red consuman servicios de la misma manera, haciendo uso de señalización SIP. A su vez, posibilita la integración de dichas redes, tales como Internet o la red de telefonía móvil.

En el contexto de la red IMS, la función de AS podría realizarse por cualquier servidor Web, entre los que podemos destacar Tomcat o GlassFish [33][30].

Sin embargo, el servidor de aplicaciones SailFin [35], que implementa la especificación JCP SIP Servlet 1.1 (JSR 289) integrándola en GlassFish, permite el desarrollo de aplicaciones convergentes con una mayor facilidad. Así mismo, sufraga la necesidad de las comunicaciones y aplicaciones multimedia requeridas por el modelo de red comentado anteriormente.

Capítulo 3

Descripción general del sistema

El presente capítulo realiza una descripción general del sistema de comercio electrónico de pago seguro, con soporte para gestión de identidad del usuario y autorización de pagos mediante terminal móvil, que se ha definido durante este proyecto.

En los siguientes apartados se definirán los requisitos funcionales del sistema completo, y se especificarán los diferentes módulos que lo componen, haciendo hincapié en las relaciones y dependencias existentes entre ellos.

Al finalizar el capítulo, el lector tendrá una idea más clara del propósito del proyecto y de la arquitectura del mismo, para así poder abordar mejor los siguientes capítulos, en los que el diseño de los diferentes módulos se explica más detalladamente.

3.1. Requisitos funcionales

El sistema desarrollado tiene como objetivo la provisión de un servicio de pago seguro en un comercio electrónico a través de una aplicación convergente. Para ello, se proporciona al sistema soporte a la verificación de los pagos mediante un terminal IMS, para así aumentar la seguridad en las transacciones y el grado de confianza de los usuarios en dicho sistema.

El riesgo que conlleva que los usuarios proporcionen su información de identidad hace que la gestión de identidad sea necesaria en el sistema. Por otro lado, la peligrosidad que acarrea el uso de tarjetas de crédito para compras a través de la Web hace que se plantee el desarrollo de una pasarela de pago para proteger las transacciones económicas entre el usuario y el comercio electrónico.

Los principales requisitos a cumplir por el sistema son:

- Provisión del servicio de comercio electrónico con autorización del pago a través de la red IMS con la garantía de los niveles de seguridad requeridos para tal efecto. El sistema deberá disponer de diversos módulos de identificación/autenticación para verificar la identidad del usuario:
    - Módulo de gestión de identidad: Para identificar al usuario en el comercio electrónico y acceder a sus opciones de pago sin necesidad de solicitar datos innecesarios.
    - Módulo de autenticación mediante terminal móvil: Para verificar el pago mediante un módulo que envíe un mensaje al terminal del usuario y verifique la respuesta enviada por éste.
- Provisión del servicio de pasarela de pago asociada al servicio anterior, que valide la información proporcionada por el comercio electrónico y gestione el módulo de

autenticación por terminal IMS para confirmar el pago, procesando la respuesta recibida del mismo.

- El sistema deberá contar con una aplicación en el terminal del cliente, y así facilitarle la validación de sus transacciones bancarias con el comercio.

  Dicha aplicación será un servicio de mensajería automática, y soportará firma digital RSA para evitar ataques de suplantación de identidad del usuario (aunque también podría haberse protegido mediante una aplicación EMV contenida en la tarjeta SIM del terminal).

- El sistema debe contar con interoperabilidad entre el sistema de gestión de identidad y el módulo de autenticación, ya que la información almacenada en el perfil de identidad usuario podrá ser utilizada posteriormente por dicho módulo para realizar el proceso de verificación del pago.

3.2. Arquitectura

A continuación se detallan los principales módulos que conforman el sistema, indicando la función que desempeña cada uno de ellos y las dependencias existentes con el resto:

- Módulo de gestión de identidad: Entidad de confianza que estará destinada a emitir, mantener y gestionar la información de identidad relativa a un usuario. Posee la infraestructura para la autenticación e implementa la funcionalidad de creación, modificación y/o borrado de perfiles de usuario, almacenamiento de atributos y recuperación de los mismos. Estará formado por las siguientes entidades:

  - Proveedor de Identidad: Provee la infraestructura necesaria para gestionar la identidad digital del usuario.

  - Aplicación de comercio electrónico: Realiza las funciones de proveedor de servicio y de Relying party en el módulo de gestión de identidad. Solicita la información de identidad del usuario, la interpreta y redirige al comprador a su módulo de pago (obtenido a partir de la identidad del usuario). Una vez autorizada la compra por el usuario, recibe la respuesta de dicho módulo de pago y detalla si la compra se ha realizado con éxito o no.

- Módulo de verificación de pago seguro a través de terminal IMS: Entidad orientada a proporcionar un mecanismo de pago seguro (a través de dos canales de comunicación) para la aplicación de comercio electrónico asociada al módulo de gestión de identidad comentado anteriormente. Para ello desplegará una aplicación de pasarela de pago que protegerá la información del pago realizado y verificará la transacción a través del terminal IMS de dicho usuario. Estará formado por las siguientes entidades:

  - Módulo de autenticación mediante mensajería SIP: Entidad encargada de controlar el acceso a un servicio, el pago en un comercio electrónico en este caso particular, mediante intercambio de mensajes con el usuario final,

enviando un documento XML con la información de la compra al terminal IMS del comprador y verificando la respuesta firmada devuelta por el mismo.

- Aplicación de pasarela de pago: Aplicación de servicio encargada de acceder a la información del perfil de identidad del usuario, transmitir los datos del usuario requeridos por el módulo de autentificación por móvil y verificar el estado de la operación una vez finalizado el proceso.

- Aplicación cliente: Aplicación que recibirá la información del pago proveniente del módulo de autenticación y presentará un interfaz gráfico al usuario para validar la compra, enviando un mensaje de respuesta firmado digitalmente al mismo módulo.

En la Figura 3.1 se muestran los módulos presentados anteriormente y se reflejan las principales relaciones existentes entre ellos.

En primer lugar podemos observar como los proveedores de servicio, el comercio electrónico y la pasarela de pago, se encuentran alojados en un servidor de aplicaciones, siendo capaces de comunicarse entre sí para proporcionar la funcionalidad descrita en 3.1. Se ha seleccionado como servidor de aplicaciones Sailfin [35], extensión de servidores del tipo Glassfish [33] con soporte para manejo de SIP (Session Initiation Protocol), ya que uno de los objetivos del proyecto es integrar las aplicaciones de servicio en una red IMS.

El módulo de gestión de identidad desplegado en el sistema estará compuesto por un proveedor de servicio (en nuestro caso la aplicación de comercio electrónico), un proveedor de identidad y un RP que gestionará las peticiones de autenticación realizadas a dicho IdP.

El módulo de gestión de identidad proporcionará acceso al perfil de identidad del usuario y a los atributos almacenados en el mismo. Por ello, de forma paralela al desarrollo del sistema especificado en este texto, se desplegará un proveedor de identidad en el que se crearán determinados perfiles de usuario para poder probar el funcionamiento del mismo.

La aplicación de comercio electrónico soportará el registro del cliente mediante su identificador ID (su XRI) y posteriormente obtendrá la información de pago almacenada en el perfil de dicho cliente para redirigirle a su pasarela de pago.

La aplicación de pasarela de pago hará uso de un módulo de autenticación mediante mensajería SIP, proporcionándole los datos necesarios para que se comunique con el usuario a través de su terminal IMS para que éste confirme la compra.

Ambos módulos, de gestión de identidad y de autenticación mediante mensajería, deberán ser interoperables. La interoperabilidad consistirá en que el módulo de autenticación será capaz de obtener los atributos que requiera del perfil de identidad asociado al usuario. Para ello, podrá acceder a la información del módulo de gestión de identidad.

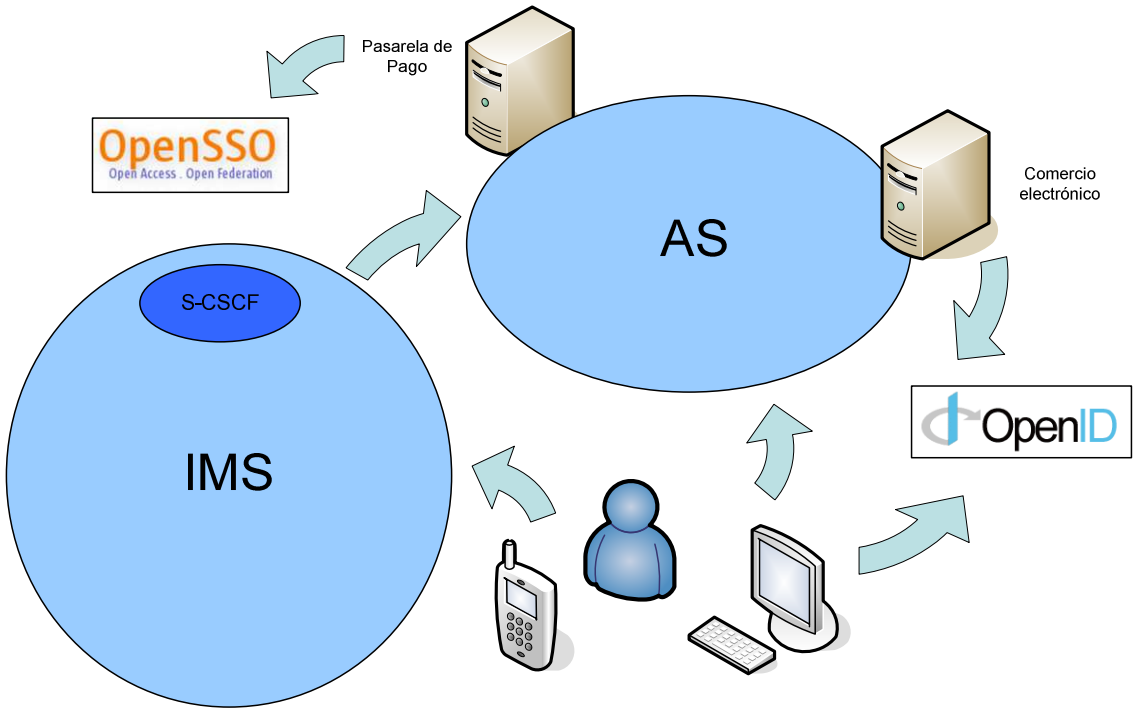

Figura 3. 1: Diagrama del sistema de autenticación mediante teléfono móvil para provisión de servicio de pago seguro en comercio electrónico

La aplicación cliente desarrollada, una vez haya registrado al cliente a la red IMS, interactuará con el módulo de autenticación a través de la pasarela de pago, intercambiando mensajes SIP para confirmar el pago de la compra realizada. IMS localizará las aplicaciones de servicio mediante el S-CSCF (Server Call Session Control Function), que almacena la información de la situación del servidor de aplicaciones en la red.

3.3. Selección de entorno

En este apartado se detallarán las principales herramientas seleccionadas para el desarrollo de los diferentes módulos especificados en 3.2, y que conforman el sistema completo.

En primer lugar, el sistema, como hemos comentado en el apartado anterior, deberá contar con un sistema de gestión de identidad. Concretamente, la aplicación de comercio electrónico será la que realice la función de proveedor de servicio para dicho sistema de gestión de identidad.

Se ha seleccionado OpenID como especificación para la gestión de identidad. La gran versatilidad de OpenID, como se comentaba en 2.3.6, el crecimiento experimentado por dicha tecnología y la existencia de gran variedad de implementaciones de la especificación han fundamentado su elección frente a SAML, SXIP o Infocards.

OpenID implementa gestión de identidad centrada en el usuario o “user-centric”, permitiendo que dicho usuario tenga control total sobre la información que proporciona a las

diferentes aplicaciones que la requieran. Mediante dicho procedimiento, se pretende evitar que el usuario tenga que proporcionar más información de la necesaria al acceder a un determinado servicio.

Por otro lado, en OpenID la gestión de identidad es distribuida, no limitándose el acceso a las aplicaciones del círculo de confianza. El usuario posee una cuenta única de OpenID, con su perfil de usuario, a la que irá asociando las diferentes aplicaciones a las que proporciona acceso sin necesidad de volver a introducir su nombre de usuario y su contraseña (soporta SSO). Un inconveniente del uso de OpenID deriva de que el uso de gran número de páginas web ocasiona que los servicios y las operaciones disponibles sobre los mismos sean menores.

Como se comentó en 3.2, el sistema desarrollado, deberá contar con un proveedor de identidad OpenID para implementar el módulo de gestión de identidad descrito. Para desplegar dicho IdP se ha seleccionado OpenSSO.

OpenSSO, como se comentó en 2.3.5, es un producto para la gestión de identidad que permite integrar diferentes especificaciones de gestión de identidad, protocolos de intercambio de información y módulos de autorización/autenticación. Para ello actúa bajo el concepto de meta-IDP, cuya estructura se detalla en la Figura 3.2:

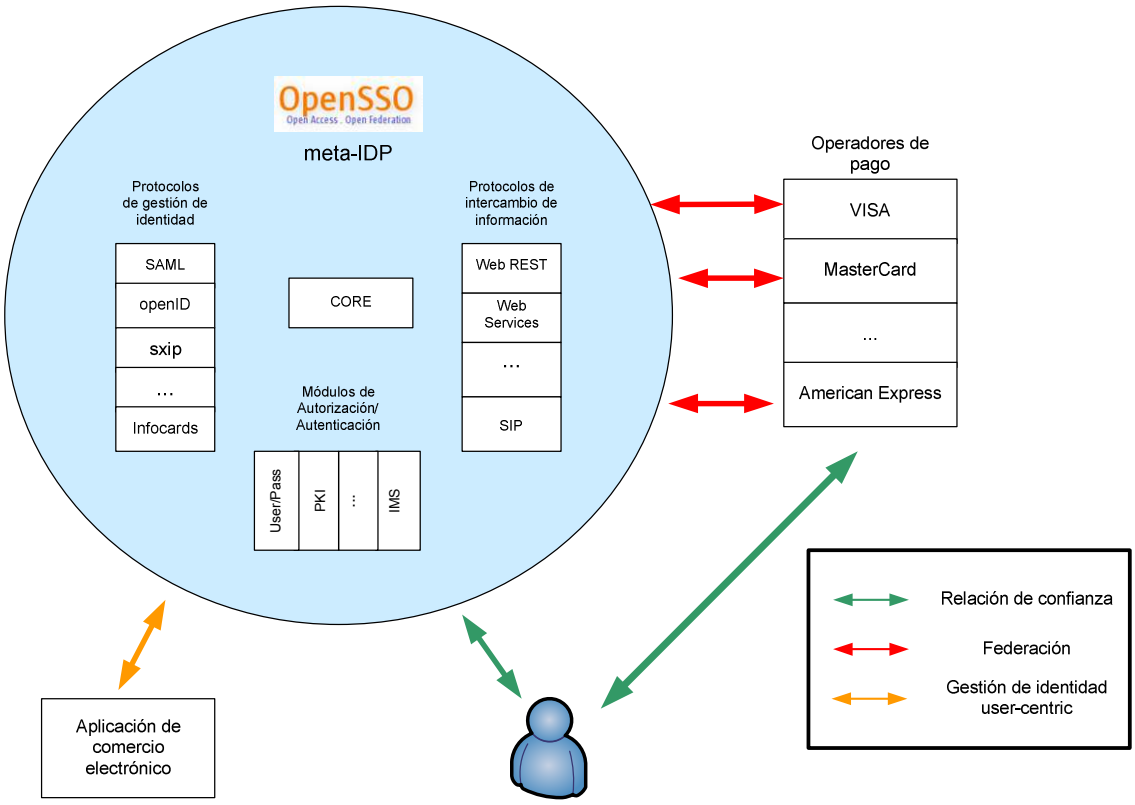

Figura 3. 2: Arquitectura del sistema de autenticación mediante terminal IMS para la provisión de servicio de pago seguro en comercio electrónico.

OpenSSO ofrece la posibilidad de realizar gestión de identidad haciendo uso de una gran variedad de especificaciones, de las cuales se ha seleccionado OpenID para la realización del presente proyecto. Por otro lado, OpenSSO proporciona una gran variedad de mecanismos de autenticación para controlar el acceso a servicios, tales como par nombre de usuario/contraseña o PKI. Además, existe la posibilidad de definir nuevos módulos de autenticación personalizados, lo que supone un marco idóneo para el desarrollo del módulo de autenticación mediante mensajería SIP requerido por sistema descrito en este proyecto.

En OpenSSO, la sesión establecida tiene como ámbito el círculo de confianza. Por otro lado, el usuario posee su perfil en cada una de las aplicaciones que conforman el círculo de confianza, además de poseer diferentes identificadores y contraseñas de acceso al mismo, y debe ser él quien asocie los perfiles para establecer sesiones SSO.

Las librerías de desarrollo de OpenSSO proporcionan la posibilidad de recuperar a partir de las cookies del navegador las sesiones SSO establecidas con anterioridad en el sistema. La información de la sesión SSO se obtiene en forma de "token", mediante el cual las diferentes aplicaciones pueden acceder a la información de identidad asociada al usuario.

En cuanto al usuario final, podemos distinguir que el uso de OpenSSO favorece no solo al cliente, sino que las empresas proveedoras de servicios también se benefician de la posibilidad de ser accedidas por parte de gran cantidad de usuarios pertenecientes a su círculo de confianza.

3.4. Funcionamiento del sistema

En primer lugar, y con la ayuda de la Figura 3.2, detallamos el funcionamiento general del sistema de pago seguro en comercio electrónico con autorización de la compra a través de una red IMS.

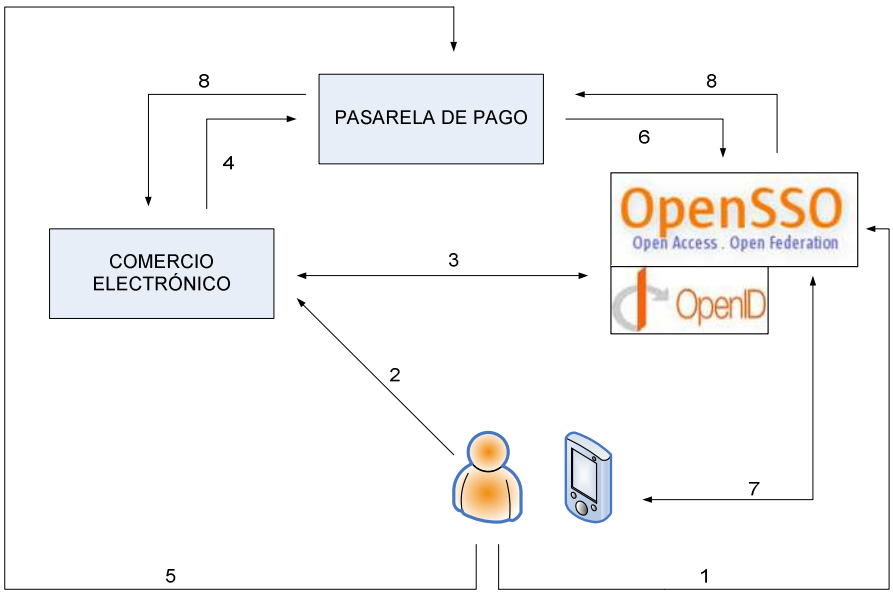

Figura 3.2: Diagrama de funcionamiento del sistema

Previamente a la puesta en marcha del sistema desplegado, deberán realizarse algunas configuraciones:

- El usuario deberá haberse registrado previamente en el proveedor de identidad OpenID y haber cumplimentado su perfil de usuario. Esto se debe a que tanto el alias del usuario, su pasarela de pago o la información de su identidad pública en IMS (IMPU) son obtenidas de dicho perfil.
- El usuario debe haber conectado su terminal IMS y haberse registrado a través de él a la red IMS con la IMPU que tenga activada en su perfil OpenSSO.
- El usuario debe haberse creado un perfil en el servicio de mensajería.
- El usuario debe activar el servicio de mensajería previamente a aceptar los términos de la compra en la pasarela de pago.

A continuación se detallan las acciones realizadas en los diferentes pasos del diagrama mostrado en la figura anterior:

1. El usuario se registra en OpenSSO y rellena su perfil de usuario. Además, registra su terminal IMS en la red y activa el servicio de mensajería en el mismo.
2. El usuario selecciona la compra deseada.
3. El comercio electrónico identifica al usuario mediante OpenID.
4. El comercio electrónico redirige al usuario a su pasarela de pago.
5. El usuario acepta los términos de la transacción.
6. La pasarela de pago activa el mecanismo de autenticación por terminal móvil en OpenSSO.
7. Módulo de autenticación y terminal IMS intercambian mensajes de validación, requiriendo interacción con el usuario.
8. La pasarela de pago recibe la confirmación del usuario y el estado de la autenticación, quedando validada o denegada la transacción comercial. Redirige al usuario de vuelta al comercio electrónico.

En los siguientes capítulos se completará la información aquí suministrada, diferenciando las características de cada uno de los módulos que componen el sistema y especificando la función que desempeñan en el mismo.

Capítulo 4

Módulo de gestión de identidad

En el siguiente capítulo se detalla el funcionamiento del módulo de gestión de identidad desarrollado en el sistema desplegado en el presente proyecto.

Como se detalló en la sección 3.3, el hecho de que los usuarios deban registrarse para acceder a servicios es cada vez más común hoy en día. Este hecho, en gran número de ocasiones, va ligado a la cumplimentación de cuestionarios en los que el cliente debe proporcionar mucha más información de la necesaria para tener acceso al servicio que el vendedor le ofrece.

Para solucionar dicho problema, se define el módulo de gestión de identidad desarrollado según la especificación OpenID [24]. Para ello, se ha hecho uso de la extensión que proporciona OpenSSO para la implementación de un proveedor de identidad OpenID.

Por otro lado, se ha implementado una aplicación de comercio electrónico que actuará como proveedor de servicio asociado al IdP OpenID. Además, dicha aplicación aloja el RP que gestiona las peticiones de autenticación entre el SP y el IdP mencionados anteriormente.

4.1. Arquitectura del módulo de gestión de identidad

El módulo de gestión de identidad presentará la arquitectura mostrada en la Figura 4.1. A continuación se detallan los principales elementos que conforman dicho sistema:

- Proveedor de servicio (SP): Solicita información de la identidad asociada al usuario al proveedor de identidad. Una vez identificado satisfactoriamente el usuario, le proporciona acceso al servicio al que deseaba acceder.
- Proveedor de identidad (IdP): Gestiona la información de identidad del usuario, almacenada en el denominado perfil de identidad. Recibe las peticiones de autenticación y de intercambio de atributos y genera respuestas a las mismas.
- Relying party (RP): Actúa como intermediario entre el SP y el IdP, generando las peticiones de autenticación y procesando las respuestas devueltas por el proveedor de identidad.

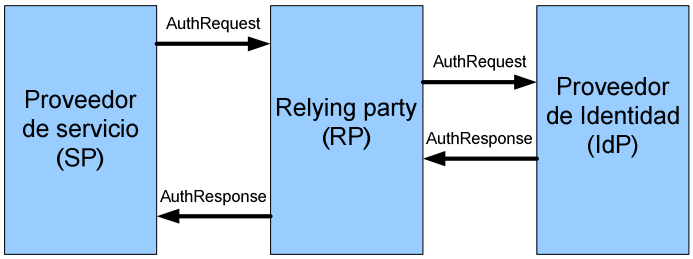

Figura 4.1: Arquitectura del módulo de gestión de identidad

Para el despliegue del módulo de gestión de identidad comentado anteriormente, se ha hecho uso de OpenSSO. Concretamente, se ha utilizado la implementación de la especificación OpenID para la gestión de identidad integrada en OpenSSO que permite desplegar un IdP.

Por otro lado, se ha implementado una aplicación de comercio electrónico que realiza las funciones de SP y de RP frente al proveedor de identidad desplegado.

La siguiente figura muestra el despliegue del módulo de gestión de identidad dentro del presente proyecto:

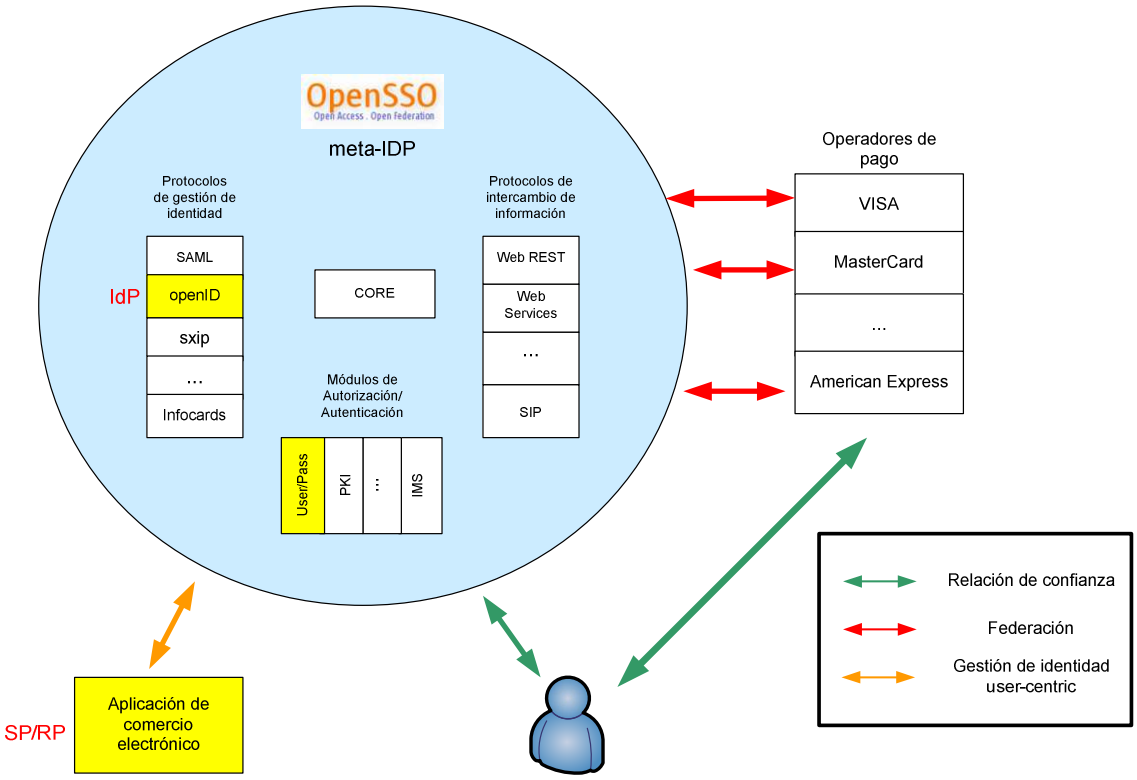

Figura 4.2: Funcionalidad del módulo de gestión de identidad en el sistema

Como podemos observar en la figura anterior, el sistema de gestión de identidad tendrá su IdP integrado en OpenSSO. A su vez, se hará uso del módulo de autenticación mediante nombre de usuario/contraseña disponible en dicho producto. Fuera de OpenSSO, se encontrará la aplicación de comercio electrónico que se viene comentando, que realizará las funciones de proveedor de servicio y de RP de dicho módulo de gestión de identidad.

4.2. Funcionalidad del módulo de gestión de Identidad

La principal función del módulo desplegado es gestionar la identidad de los usuarios que acceden a una aplicación de comercio electrónico. A su vez, se protegen los datos de dichos usuarios, evitando que tengan que proporcionar mayor cantidad de información que la estrictamente necesaria.

El protocolo de funcionamiento se divide en dos fases: iniciación de la compra y finalización de la compra.

- Iniciación de la compra:

1. El usuario debe estar previamente registrado en el proveedor de identidad OpenID.
2. El usuario selecciona el producto deseado en el comercio electrónico.
3. El comercio electrónico solicita al usuario que introduzca su identificador OpenID y le indica la información que necesita obtener de su perfil.
4. El usuario introduce su identificador.
5. Se activa el protocolo de gestión de identidad por OpenID.
6. El comercio electrónico recibe la respuesta de identificación en el RP OpenID.
7. En caso de identificación satisfactoria, se redirige al usuario a su pasarela de pago, cuya dirección ha sido obtenida en uno de los atributos solicitados al OP de OpenID y se envía en la petición el importe de la compra y una descripción de la misma.

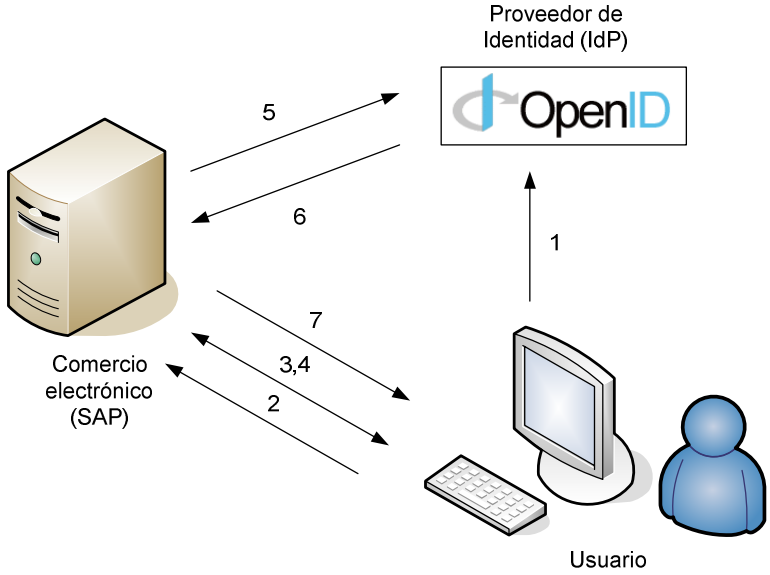

Figura 4.3: Diagrama de funcionamiento de la aplicación del comercio electrónico

- Finalización de la compra:

Dependiendo de si el pago de la compra ha sido efectuado satisfactoriamente o ha ocurrido algún error, la pasarela de pago redirigirá al usuario a la página de confirmación de pago realizado o de error, respectivamente.

Profundizando en el funcionamiento del protocolo de gestión de identidad OpenID [36], enunciado en la lista anterior (concretamente en el punto 5), se enumeran las acciones que componen dicho proceso:

1. El usuario solicita el inicio del proceso de autenticación, introduciendo su identificador OpenID (una XRI o una URL), en la web del comercio electrónico. Dicho identificador es remitido al Relying Party (RP) para comenzar el proceso.

2. Tras normalizar el identificador proporcionado por el usuario, el RP realiza un proceso de descubrimiento sobre el mismo para determinar la dirección de autenticación del proveedor de identidad (OP) asociado a dicho usuario.

   Cabe denotar que el identificador proporcionado por el usuario debe ser un identificador OP para su correcta interpretación por el protocolo y posibilitar el uso de extensiones. Dependiendo de que el identificador remitido sea una URL o un XRI se realizará el descubrimiento mediante Yadis o por descubrimiento XRI, respectivamente.

3. El RP y el OP establecen una asociación de secreto compartido usando el método para intercambio de claves Diffie-Hellman. El OP hará uso de dicha asociación para firmar los mensajes y el RP para verificarlos, eliminando la necesidad de realizar peticiones de verificación de la firma tras cada proceso de autenticación.

4. El RP redirige al usuario al OP mediante una Authentication Request (Petición de autenticación). En el caso particular que nos ocupa, al necesitar del intercambio de atributos, se añadirá una extensión a la petición de autenticación remitida, con una FetchRequest en la que se adjuntan los mismos.

5. El OP autentica al usuario mediante nombre de usuario y contraseña. Posteriormente comprueba si el usuario está autorizado para autenticarse mediante OpenID y, en caso afirmativo, redirige al usuario a una página dónde le permite seleccionar los atributos de su perfil que desea proporcionar al SP.

6. El OP redirige al usuario de vuelta al RP con, dependiendo del caso, un aserto probando que la autenticación ha sido aprobada (Positive Assertion), o un mensaje de fallo en la autenticación (Negative Assertion).

7. El RP verifica la información recibida del OP:

   - Comprobación de la URL de retorno.
   - Verificación de la información descubierta.
   - Comprobación del reto emitido.
   - Verificación de firma.

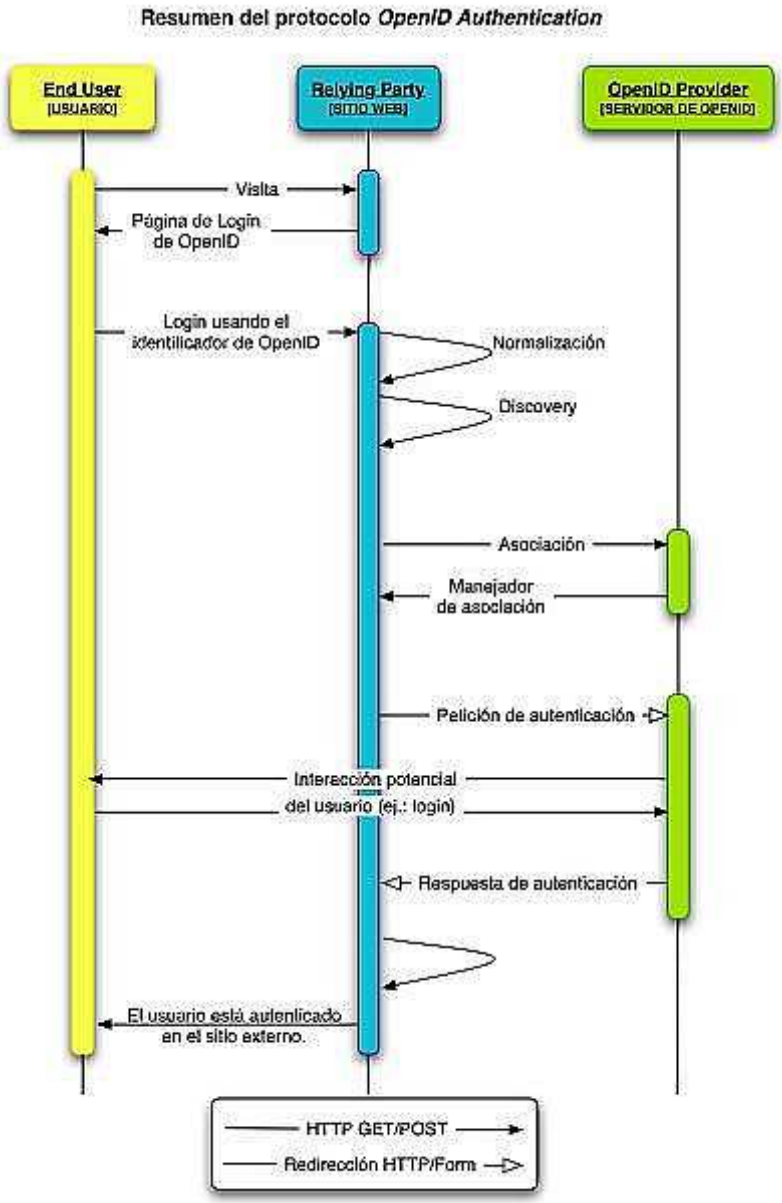

Figura 4.4: Diagrama de peticiones/respuestas entre entidades participantes en el proceso OpenID Authentication. Obtenida de [38].

4.3. Despliegue del proveedor de Identidad

El sistema de gestión de identidad desplegado en el sistema permitirá el desempeño de las siguientes especificaciones de OpenID:

- Autenticación mediante OpenID 2.0. : A través de esta especificación, somos capaces de realizar una petición de autenticación frente al proveedor de identidad y recibir la respuesta posteriormente en nuestro proveedor de servicio para realizar la verificación de la misma.

- Intercambio de atributos OpenID 1.0. : Mediante la inclusión de una extensión a la petición de autenticación se solicita el intercambio de atributos pertenecientes al perfil del usuario que se desea validar.

Para poner en funcionamiento el sistema de gestión de identidad, es necesario desplegar dicho proveedor de identidad OpenID (OP, OpenID Provider en inglés) para poder acceder al perfil del usuario desde el RP alojado en la aplicación del comercio electrónico. A pesar de estar integrado en OpenSSO, el OP de OpenID constituirá una instancia diferente en el servidor de aplicaciones. Para la correcta puesta en marcha del sistema, deberemos realizar ciertas configuraciones tanto en el módulo OpenID como en OpenSSO, que pueden consultarse en el Apéndice B.

4.4. Implementación del proveedor de servicio: Aplicación de comercio electrónico

4.4.1 Proveedor de servicio

El servicio de comercio electrónico se encargará de proporcionar una interfaz Web al usuario para la realización de su compra y la posibilidad de identificarse mediante su identificador OpenID, por ello, el RP perteneciente al módulo de gestión de identidad OpenID se encuentra alojado en la aplicación.

Para el despliegue del sistema de gestión de identidad mediante OpenID se ha seleccionado la librería OpenID4Java [39], desarrollada en Java, como su propio nombre indica, debido a su facilidad de uso y a que el módulo de OpenID implementado por OpenSSO se basa en dicha especificación.

OpenID4Java proporciona soporte para los siguientes servicios:

- Autenticación OpenID 1.1.
- Autenticación OpenID 2.0.
- Intercambio de atributos OpenID 1.0.
- Registro simple OpenID 1.0 y 1.1.
- Extensión para políticas de provisión de autenticación OpenID 1.0.
- Infocards OpenID 1.0.

El proveedor de servicio implementado tan solo hará uso de las especificaciones de autenticación OpenID 2.0 y de intercambio de atributos.

La aplicación J2EE que implementa el comercio electrónico consta de un HTTP Servlet, y de varios JSP's que interactúan con él. A continuación se describen las principales características de las entidades que se alojan en la aplicación descrita.

Entidad	Descripción
Welcome	Es la página de bienvenida del comercio electrónico. Ofrece al usuario una lista de productos determinada entre los que seleccionará uno y solicitará la compra.
Registrar	Página de registro. Para la realización de una compra en el comercio electrónico descrito, es necesario identificarse en el mismo, ya bien mediante nombre de usuario y contraseña para el propio lugar, o bien mediante OpenID. El interés del proyecto se basa en el uso de OpenID como mecanismo de identificación centralizada, por lo que el documento se focalizará en su uso.
Index	Es la página de solicitud de identificación mediante OpenID, en la que el usuario deberá introducir su identificador OpenID (XRI) y autorizar la solicitud de los atributos que el comercio necesite a su proveedor de identidad OpenID.
FormRedirection	Página de redirección al proveedor de identidad del usuario, que es descubierto mediante el proceso de descubrimiento en ConsumerServlet.
RegistradoNormal	Página de muestra de identificación satisfactoria mediante uso de nombre de usuario y contraseña. Su uso queda fuera del interés del proyecto y se ha diseñado en vista a la apariencia estética de la aplicación.
ConsumerServlet	Clase que actúa como Relying Party de la aplicación de comercio electrónico frente al proveedor de identidad OpenID. Su funcionamiento será descrito con una mayor profundidad a continuación, ya que constituye una gran parte del módulo de identificación con OpenID.
Return	Página de retorno para transacciones

	finalizadas con éxito. La pasarela de pago redirigirá al usuario a esta página en caso de que todo el proceso de compra se haya finalizado y dicha compra se haya autorizado correctamente.
Error	Página de retorno para transacciones finalizadas sin éxito. La pasarela de pago redirigirá al usuario a esta página en caso de que se haya producido un error en el proceso de compra, el usuario haya cancelado la misma o se hayan detectado irregularidades en la transacción.

Una vez presentada la estructura de la aplicación, en las Figuras 4.5 y 4.6 se muestra el diagrama funcional de la aplicación y el flujo de llamadas que se produce entre las clases que componen la aplicación a lo largo del proceso de compra.

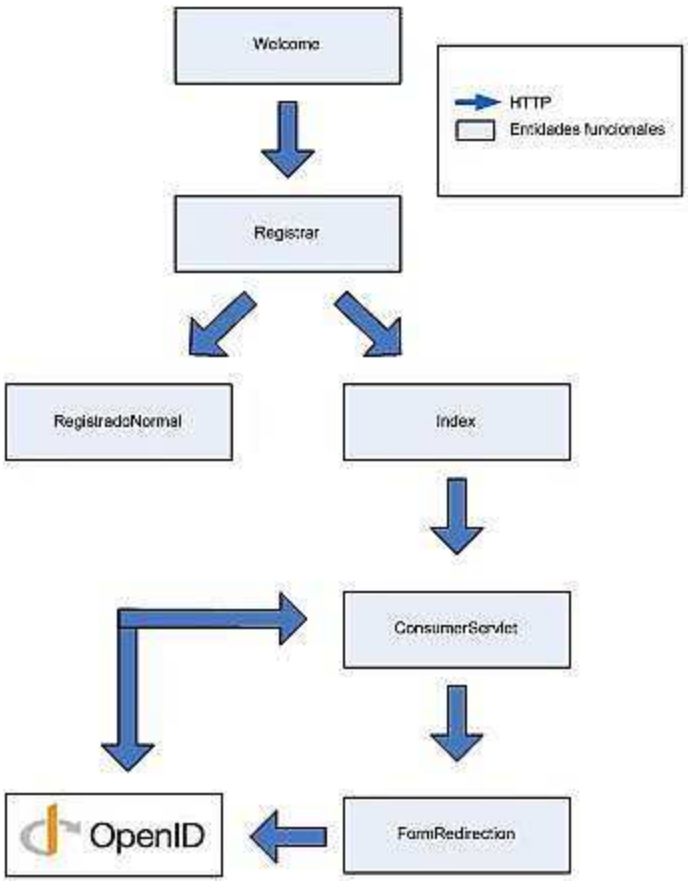

Figura 4.5: Diagrama funcional de la aplicación del comercio electrónico en la iniciación de la compra

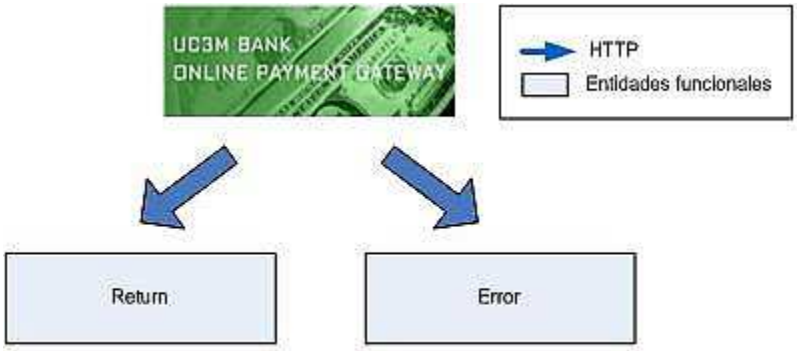

Figura 4.6: Diagrama funcionad de la aplicación del comercio electrónico en la finalización de la compra

En el proceso de iniciación de la compra, como se muestra en la Figura 4.5, el usuario accede a la página de bienvenida del comercio electrónico, llamada `Welcome.` En ella, selecciona el producto que desea comprar de la lista que se le presenta y es redirigido a la página `Registrar`.

Se solicita al usuario su identificación, permitiéndole hacerlo mediante nombre de usuario y contraseña (para redirigirle a `RegistradoNormal`, página diseñada simplemente para dotar a la aplicación de una estética más realista), o bien mediante OpenID. Para ello deberá introducir su XRI en el cuadro de texto correspondiente.

Posteriormente, el usuario será redirigido al RP, implementado mediante la clase `ConsumerServlet`, que realizará el descubrimiento del proveedor de identidad asociado al XRI del usuario. El usuario será redirigido nuevamente, esta vez a la página de su proveedor de identidad.

Una vez se encuentre en la página del proveedor de identidad, el usuario deberá iniciar sesión en el mismo, introduciendo su nombre de usuario y su contraseña. Si la operación de login finaliza con éxito, se presentará al usuario una página en la que selecciona los atributos que proporcionará al comercio electrónico. Una vez realizada su selección y proporcionada su confianza al proveedor de servicio, será redirigido de nuevo al RP.

El RP procesará la respuesta obtenida del IdP y analizará los atributos obtenidos del perfil de identidad del usuario, entre los que se encuentra la pasarela de pago que dicho usuario tiene asociada.

En este punto, habría finalizado el proceso de iniciación de la compra, y el usuario sería redirigido a su pasarela de pago para verificarse la transacción, como detallaremos en el siguiente capítulo.

Una vez la transacción se haya finalizado, la pasarela de pago redirigirá al usuario de vuelta a la aplicación de comercio electrónico. Dicho proceso lo hemos definido como finalización de la compra y, como vemos en la Figura 4.6, finalizará presentando al usuario la página `Return` o la página `Error`, dependiendo de si la operación ha finalizado con éxito o no, respectivamente.

4.4.2 Relying party

Dentro del módulo de gestión de identidad, resulta vital la función desempeñada por el Relying Party, por lo que a continuación se describe de forma más detallada el funcionamiento de la clase `ConsumerServlet`, que ofrece dicha funcionalidad frente al IdP comentado en el apartado anterior.

La función realizada por esta entidad será la de gestionar la creación de peticiones de autenticación en las que se solicite intercambio de atributos según las preferencias del proveedor de servicio. A su vez, se encargará de procesar las respuestas devueltas por el proveedor de identidad, verificando las mismas y determinando si la identificación del usuario ha sido satisfactoria o no.

Cabe destacar que la clase implementa la interfaz HTTPServlet, lo que significa que la misma recibirá peticiones Web provenientes de otras páginas y actuará en respuesta a dichas peticiones, como hemos explicado antes.

Los principales atributos de la entidad son:

`String paymentGateway`	Pasarela de pago asociada al usuario que realiza la compra.
`String description`	Descripción acreditativa de las compras realizadas en el comercio electrónico.
`String quantity`	Importe de la compra realizada.

Seguidamente se detallan las principales funciones definidas en la clase `ConsumerServlet`:

`void init (ServletConfig config)`	Inicializa el Servlet y recupera el contexto almacenado en la configuración. • config: Almacena la configuración asociada al Servlet.
`void doPost (HttpServletRequest req, HttpServletResponse resp)`	Procesa las peticiones realizadas al Servlet. Verifica si la petición es de respuesta o no y, la procesa o inicia la autenticación llamando al método authRequest, respectivamente.

	• req, resp: Par petición/respuesta asociada al Servlet.
String authRequest (String userSuppliedString, HttpServletRequest httpReq, HttpServletResponse httpResp)	A partir del identificador recibido, realiza el descubrimiento del OP asociado al mismo y crea la petición de autenticación a la que añade la extensión de intercambio de atributos (modo AX: Attribute Exchange). Finalmente redirige al usuario a su proveedor de identidad asociado, enviando su petición de autenticación asociada. • userSuppliedString: Identificador proporcionado por el usuario para su identificación por OpenID. • req, resp: Par petición/respuesta asociada al Servlet.
void processReturn (HttpServletRequest req, HttpServletResponse resp)	Procesa la respuesta proveniente del OP. Tras verificar la respuesta mediante la llamada al método `verifyResponse`, recupera el valor del atributo asociado a la dirección de la pasarela de pago del cliente y le redirige a ella. • req, resp: Par petición/respuesta asociada al Servlet.
Identifier verifyResponse(HttpServletRequest httpReq)	Realiza las diferentes comprobaciones pertinentes para verificar la autenticación del usuario: comprobación de la URL de retorno, verificación de la información descubierta, comprobación del reto emitido y verificación de la firma. • req: Petición asociada al Servlet. Devuelve el identificador asociado a la autenticación verificada.

4.5. Conclusiones

El módulo de gestión de identidad desplegado provee al sistema desarrollado en el presente proyecto de la capacidad para proteger la identidad digital asociada a los usuarios,

permitiendo que ellos tengan la última palabra sobre cuándo, a quién y qué tipo de información quieren proporcionar.

En un entorno en el que el crecimiento exponencial de las cuentas asociadas a un usuario en los diferentes proveedores de servicio, le gestión de identidad se convierte en un factor clave. Como consecuencia de su uso, el usuario podría tener acceso a diferentes proveedores de servicio con las mismas credenciales.

La creación de un perfil de usuario como el que se ha realizado en este proyecto, en el que se almacenan credenciales y atributos tan dispares como el IMPU de un usuario o la pasarela de pago asociada al mismo, dotan de una gran versatilidad al sistema de gestión de identidad y permiten su utilización en gran variedad de entornos.

Capítulo 5

Módulo de verificación de pago seguro a través de terminal IMS

En el presente capítulo se detallarán la arquitectura y el funcionamiento del módulo de verificación de pagos desarrollado en el proyecto. Dicho módulo, consistirá en una pasarela de pago asociada al usuario que hará uso de un módulo de autenticación por mensajería SIP.

El módulo de autenticación, a su vez, verificará la transacción económica asociada a la compra mediante el intercambio de mensajes SIP con una aplicación cliente alojada en el terminal IMS del usuario.

La verificación del pago se realizará a través de una red IMS, lo que dotará al sistema de una verificación a través de dos canales de comunicación: Internet e IMS. La mayor confianza existente en la tecnología móvil y la mayor robustez de la misma hacen que el sistema presente una gran seguridad para la transmisión y el intercambio de información sensible (números de cuenta o de tarjetas de crédito) entre pasarela de pago y el cliente.

5.1. Arquitectura del módulo de verificación de pago

Como se especificó en 3.2, el sistema de verificación de pago a través de red IMS consta de tres módulos diferenciados, cuya funcionalidad básica se detallará a continuación. Su funcionalidad dentro del sistema completo desplegado en el proyecto se muestra en la Figura 5.1.

- Pasarela de pago: Aplicación que proporciona al usuario el soporte para la realización de transacciones económicas seguras, verificando el pago de las compras realizadas mediante un módulo de autenticación que permite al usuario aceptar los términos de dichas transacciones a través de su terminal IMS.

  La aplicación de pasarela de pago se encuentra alojada en un servidor de aplicaciones y sirve de canal para el pago seguro. Para ello, actuará como intermediario entre el comercio electrónico y el usuario, dotando a la transacción de fiabilidad, robustez y transparencia.

- Módulo de autenticación mediante terminal IMS: Módulo de autenticación integrado en OpenSSO que permitirá a la pasarela de pago verificar la transacción con la aplicación cliente mediante el intercambio de mensajes XML firmados [40].

  La comunicación se realiza a través de la red IMS y los mensajes intercambiados son señalización SIP.

- Aplicación cliente de mensajería: Aplicación integrada en el terminal IMS del usuario que le permite verificar las transacciones realizadas en un comercio electrónico.

Una vez seleccionada la compra, y a través de la pasarela de pago, el módulo de autenticación anteriormente mencionado y la aplicación cliente intercambian mensajes de verificación para autorizar la compra.

Dicha aplicación cliente presenta un interfaz simple e intuitivo para el usuario, facilitándole el proceso de verificación del pago.

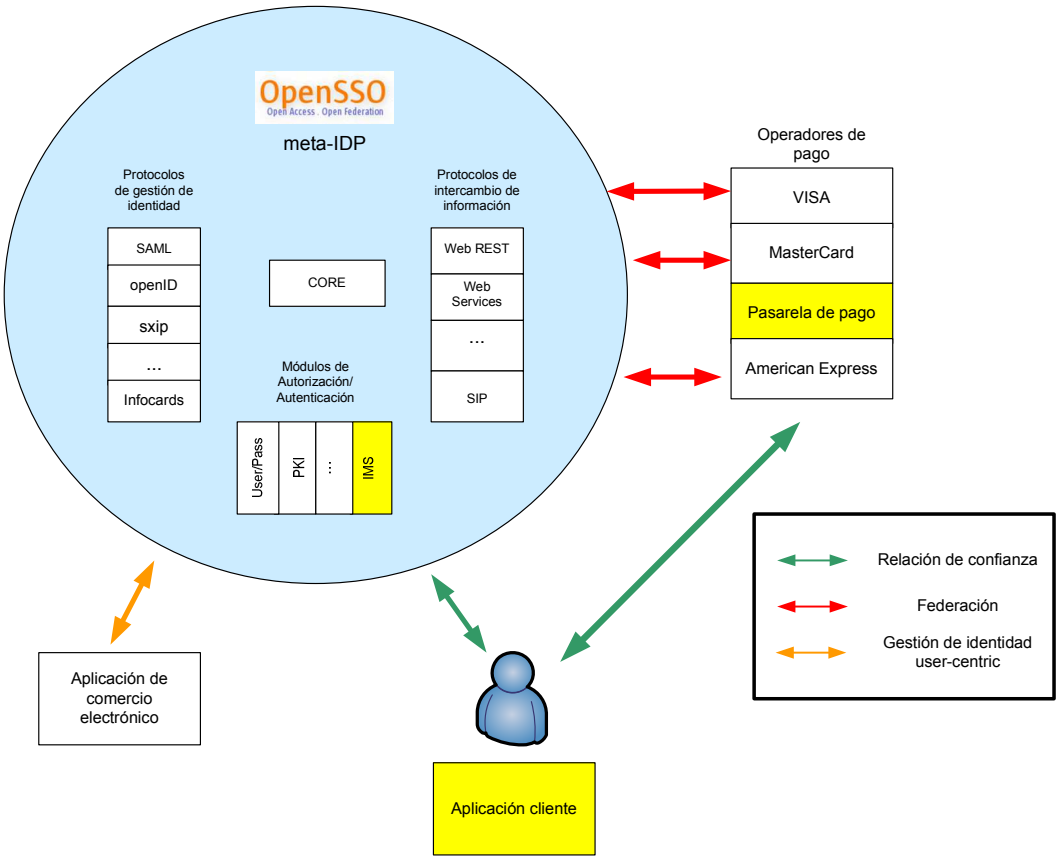

Figura 5.1: Funcionalidad del módulo de verificación de pago en el sistema

5.2. Módulo de autenticación mediante terminal IMS

5.2.1. Descripción del módulo

En el este apartado se detalla el módulo de autenticación mediante mensajería SIP desarrollado en OpenSSO, que permite al usuario verificar el pago de la compra realizada a través de la aplicación de comercio electrónico.

Como se viene repitiendo a lo largo del texto, la necesidad de un nivel de seguridad más elevado a la hora de efectuar transacciones económicas a través de la red IMS hace que surja la idea de llevar a cabo la autenticación mediante un segundo canal de comunicación: la telefonía móvil.

Se ha implementado un mecanismo personalizado para la provisión de autenticación mediante uso de terminal móvil IMS. Para ello, se ha registrado un nuevo módulo en la especificación OpenSSO [22] , como vemos en la figura 5.2.

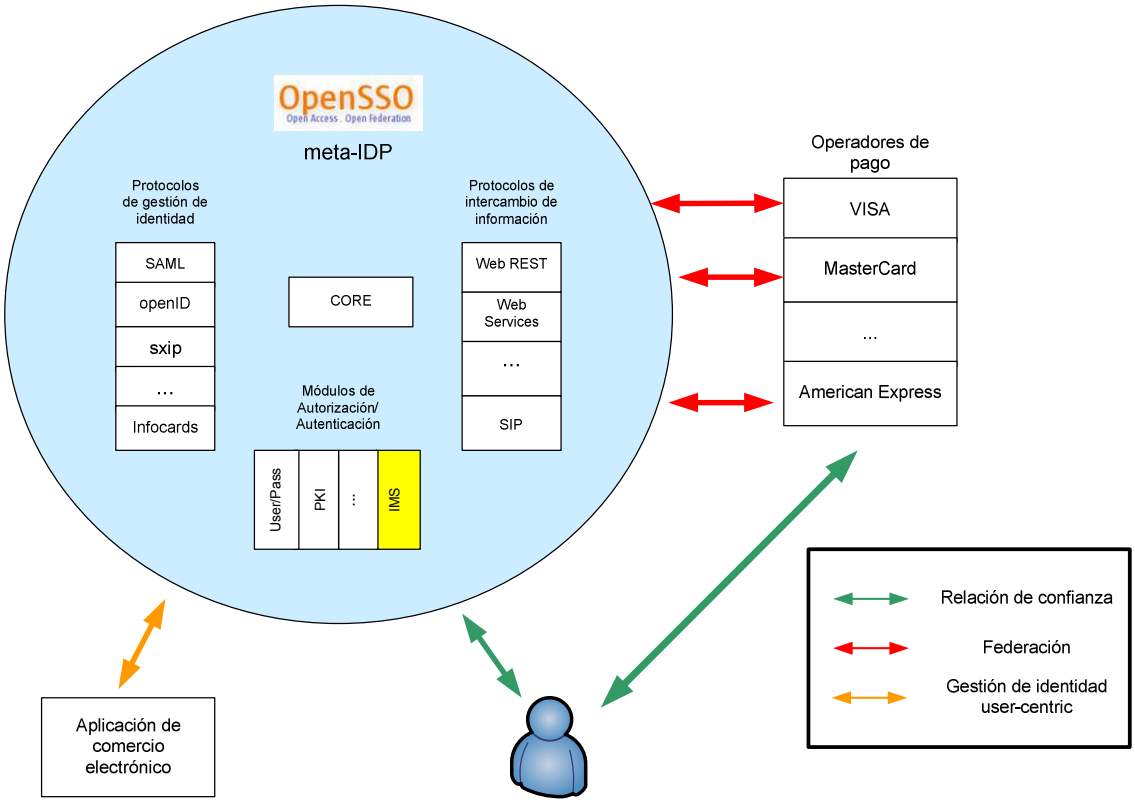

Figura 5.2: Funcionalidad del módulo de autenticación en el sistema

Antes de definir la funcionalidad del módulo, describiremos el concepto de "callback". Al iniciarse el proceso de autenticación, el módulo remite sus requisitos a la entidad que desea autenticarse. Dichos requisitos se presentan en forma de "callback", que no es más que un formato predeterminado para expresar requisitos de tipo cadena de caracteres (`NameCallback`), contraseñas (`PasswordCallback`),etc.

Los principales requisitos del módulo de autenticación son los siguientes:

- El terminal IMS del usuario. Éste debe encontrarse correctamente registrado a la red IMS y debe ser accesible al proveedor de servicio que está siendo utilizado.
- El identificador de usuario o "uid". El usuario deberá haberse identificado mediante OpenID previamente en el comercio electrónico, ya que tanto el alias como la información de su identidad pública en IMS (IMPU), son obtenidas por el módulo a partir del token generado en dicha autenticación.
- El comercio electrónico deberá remitir los siguientes atributos, necesarios para la confirmación de la compra:
  - Importe total de la compra a cobrar.
  - Descripción de la compra (no contendrá información de los productos, sino del comercio que emite la orden de cobro).

- La pasarela de pago deberá remitir los siguientes atributos, necesarios para la confirmación de la compra:
    - Módulo de la clave pública del cliente.
    - Exponente de la clave pública del cliente.

La ejecución del mecanismo de autenticación, una vez cumplidos los requisitos especificados anteriormente, sigue el siguiente protocolo (representado en la Figura 5.3):

1. La pasarela de pago recibe los callbacks solicitados por el módulo de autenticación según su configuración.
2. El módulo recibe la petición de autenticación. Comprueba si se han remitido los callbacks requeridos, en caso contrario finaliza la autenticación con error.
3. Comprueba si existe una petición anterior del usuario.
    3.1. Si existe una petición anterior, la recicla con los valores recibidos.
    3.2. Si no existe crea una nueva con los valores recibidos.
4. Generación del mensaje XML de autorización de la compra a partir de los atributos recibidos.
5. Envío del mensaje al terminal IMS del usuario.
6. Recepción de mensaje de confirmación/denegación del pago.
7. Comprobación del mensaje y establecimiento de estado de autenticación dependiendo de si es válido y según el pago se haya aceptado o denegado.
8. Una vez el usuario haya aceptado los términos de la compra en la pasarela de pago, recepción de nueva petición de autenticación en estado “en progreso”. Comprueba si se ha remitido el callback requerido, en caso contrario finaliza la autenticación con error.
9. Interpreta el callback recibido:
    9.1. Si cancela la operación, se finaliza la autenticación con error.
    9.2. Si confirma la operación, se devuelve a la pasarela el estado de la autorización, pudiendo ser positivo o negativo, dependiendo del mensaje de respuesta recibido anteriormente.

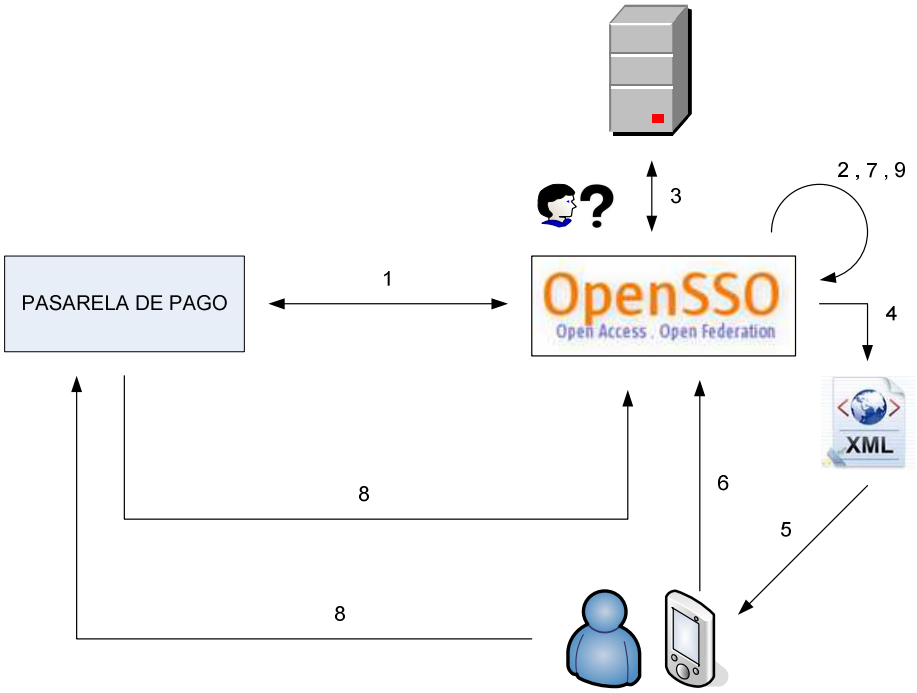

Figura 5.3: Diagrama de funcionamiento de módulo de autenticación por teléfono móvil

5.2.2. Configuración del módulo

Como se puede deducir de lo comentado anteriormente, el funcionamiento básico de un módulo de autenticación integrado en OpenSSO está íntimamente relacionado con el fichero de configuración del mismo, en el cual se encuentran definidos los diferentes estados que puede adoptar el módulo y los requisitos para llevar a cabo el proceso de autenticación en cada uno de ellos. A continuación se muestra el contenido del fichero de configuración del módulo de autenticación diseñado:

```
<ModuleProperties moduleName="AMAuthIMS" version="1.0" >
    <Callbacks length="6" order="1" timeout="120"
     header="IMS Authentication" >
        <NameCallback>
            <Prompt> User Name: </Prompt>
        </NameCallback>
        <NameCallback>
            <Prompt> SIP uri: </Prompt>
        </NameCallback>
        <NameCallback>
            <Prompt> Description: </Prompt>
        </NameCallback>
        <NameCallback>
            <Prompt> Quantity: </Prompt>
        </NameCallback>
         <NameCallback>
            <Prompt> Modulus: </Prompt>
        </NameCallback>
         <NameCallback>
            <Prompt> Exponent: </Prompt>
        </NameCallback>
    </Callbacks>
```

```
    <Callbacks length="1" order="2" timeout="120"
     header="Please, acknowledge the message and type Done or Cancel" >
        <NameCallback>
            <Prompt> Done/Cancel: </Prompt>
        </NameCallback>
     </Callbacks>
    <Callbacks length="0" order="3" timeout="120"
     header="The SIP URI provided is not reachable, please verify your data"
>
    </Callbacks>
    <Callbacks length="0" order="4" timeout="120"
     header="Please, authenticate first with other mechanism" >
    </Callbacks>
</ModuleProperties>
```

El módulo desarrollado presenta dos estados principales, para cada uno de los cuales se especifican los requisitos (en forma de callback), que el módulo solicitará al usuario:

- Estado 1: Estado en el que se encuentra el módulo al iniciarse el mecanismo de autenticación y que se inicia al aceptar los términos de la compra el usuario en la pasarela de pago. Sus requerimientos son del tipo NameCallback ya que contienen información propensa a ser utilizada y será necesario acceder a ella:
    - Uid: Identificador del usuario.
    - IMPU: Uri del terminal SIP asociado al usuario.
    - Description: Descripción de la transacción que está siendo validada.
    - Quantity: Importe de la transacción que está siendo validada.
    - Modulus: Módulo de la clave pública asociada al usuario.
    - Exponent: Exponente de la clave pública asociada al usuario.
- Estado 2: Estado "en progreso", en el cual se encuentra el módulo al iniciarse el intercambio de mensajes con el terminal IMS del cliente y a la espera de la llegada de la confirmación de éste. Su requerimiento es del tipo NameCallback ya que contiene información propensa a ser utilizada y será necesario acceder a ella:
    - Done/Cancel: Parámetro que indica si el usuario ha pulsado la tecla "Done" o la tecla "Cancel" en la pasarela de pago, validando el mensaje enviado desde su terminal, o anulando la transacción en curso, respectivamente.

Cabe destacar que se definieron dos estados adicionales que no han sido utilizados, orientados a expandir la funcionalidad del módulo en futuras actualizaciones.

5.2.3. Desarrollo de aplicaciones convergentes

Como se comentó anteriormente en este proyecto, la provisión de un servicio de verificación de pago seguro a través de una red IMS es uno de los grandes objetivos perseguidos.

Para tal fin, es necesario desarrollar una aplicación convergente que proporcione comunicación con el terminal IMS a la hora de verificar los pagos. Ése es precisamente el objetivo del módulo de autenticación descrito en el presente apartado.

El módulo de autenticación recibirá una petición de autenticación por parte de la pasarela de pago a través de Internet (HTTP), para posteriormente realizar un intercambio de mensajes (SIP) con el terminal móvil del usuario a través de IMS y finalmente devolver el estado del proceso de autenticación a la pasarela de pago de nuevo mediante HTTP.

De manera similar al concepto de Servlet HTTP descrito en 4.4, aparece el de Servlet SIP, que consistirá en una clase Java que:

- Podrá funcionar como UA (User Agent), generando peticiones SIP y enviándolas a sus correspondientes destinatarios.
- Podrá funcionar como AS (Application Server) recibiendo respuestas SIP de una determinada entidad o aplicación, procesando las mismas y realizando las acciones correspondientes en función de dicho procesamiento. Para tal fin implementa entre otras, las siguientes funciones, que son ejecutadas dependiendo de la naturaleza de la respuesta recibida:

doInvite	Se ejecuta al recibir un mensaje SIP de tipo INVITE.
doAck	Se ejecuta al recibir un mensaje SIP de tipo ACK.
doOptions	Se ejecuta al recibir un mensaje SIP de tipo OPTIONS.
doBye	Se ejecuta al recibir un mensaje SIP de tipo BYE.
doCancel	Se ejecuta al recibir un mensaje SIP de tipo CANCEL.
doSubscribe	Se ejecuta al recibir un mensaje SIP de tipo SUBSCRIBE.
doNotify	Se ejecuta al recibir un mensaje SIP de tipo NOTIFY.
doMessage	Se ejecuta al recibir un mensaje SIP de

	tipo MESSAGE.
doInfo	Se ejecuta al recibir un mensaje SIP de tipo INFO.
doPrack	Se ejecuta al recibir un mensaje SIP de tipo PRACK.

Analizando los requisitos de nuestra aplicación, vemos que nuestro módulo de autenticación deberá prestar ambas funcionalidades propias de los Servlets SIP, actuando como UA al iniciarse el proceso de verificación del pago (al enviar el mensaje de verificación al terminal IMS del usuario) y como AS al finalizarse dicha verificación (al recibir el mensaje de confirmación proveniente del terminal IMS del usuario).

A continuación se comentan dos de los principales elementos asociados a un SIPServlet y que permiten recuperar las sesiones SIP asociadas a un determinado servlet.

La entidad `SipApplicationSession` actuará como almacenamiento de la información de sesión de aplicación y proporcionará acceso a las sesiones SIP y HTTP establecidas en la misma.

Por otro lado, la entidad `SipContext` actuará como contenedor de la información de la aplicación convergente desarrollada. Posteriormente podrá recuperarse dicho contexto y con él, la información contenida en él.

Ambos elementos pueden obtenerse mediante JNDI (Java Naming and Directory Interface), interfaz que provee la plataforma Java para la localización de servicios mediante un directorio de identificadores.

Debido a que el módulo de autenticación desplegado debe presentar las funcionalidades de UA y AS por separado, deberá instanciarse dos veces. Este hecho hace que la sesión SIP establecida no pueda recuperarse de manera convencional a partir de los elementos `SipApplicationSession` y `SipContext` comentados**.**

Para solucionar el problema se ha hecho uso de un conjunto de clases de persistencia, consistentes en una base de datos que almacena las peticiones realizadas por los usuarios que acceden a la aplicación de comercio electrónico. El proceso de autenticación es el siguiente:

Una vez iniciada la verificación de la transacción por la pasarela de pago, el módulo de autenticación registra la petición realizada e instancia el Servlet como UA para el envío del mensaje de confirmación al terminal IMS del usuario.

Tras iniciarse el proceso de autenticación y enviarse el mensaje de verificación de pago al terminal del usuario, la respuesta podrá tardar en llegar un determinado tiempo (el tiempo que tarde en llegar el mensaje al usuario más el que tarde dicho usuario en aceptar la transacción).

Por ello, una vez el usuario acepte los términos de compra y envíe de vuelta el mensaje de confirmación de la compra, el Servlet instanciado como AS recibirá dicho mensaje. Para asociarlo a la sesión SIP establecida anteriormente obtendrá su valor de las clases de persistencia, en las que hará una búsqueda a partir del token asociado al mensaje recibido.

5.2.4. Implementación del módulo

La estructura funcional del módulo de autenticación en cuanto a diagrama de clases se muestra en la Figura 5.5.

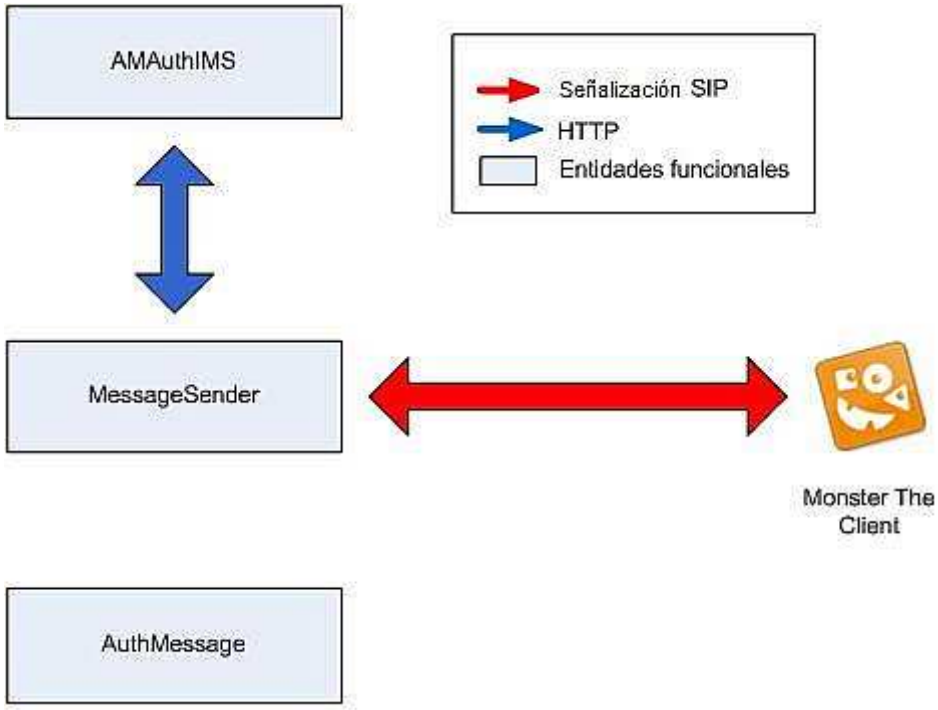

Figura 5.5: Diagrama funcional del módulo de autenticación mediante teléfono móvil

Como podemos ver en la figura anterior, se utilizan simultáneamente dos canales de comunicación, uno para la navegación por las páginas Web y otro para el intercambio de mensajes SIP.

Por otro lado, la entidad `AuthMessage` no se comunica con el resto de entidades. Esto es debido a que la función de dicha clase será establecer el estándar para los mensajes intercambiados entre el módulo de autenticación (a través de la entidad `MessageSender`) y la aplicación cliente.

La entidad `AMAuthIMS` es la encargada de gestionar el módulo de autenticación, recibiendo las peticiones de autenticación provenientes de la aplicación de pasarela de pago. Una vez recibida la petición, se comunicará con la entidad `MessageSender` para iniciar el proceso de verificación del pago.

Finalmente, la entidad `MessageSender`, será la encargada de generar los mensajes de verificación con destino a la aplicación cliente y procesar posteriormente las respuestas recibidas de dicha aplicación. En función de la respuesta recibida, validará la transacción o no,

y comunicará el resultado a la pasarela de pago para que ésta se comunique con el comercio electrónico y finalice el proceso de compra.

A continuación comentaremos más detalladamente las entidades que forman el módulo y las configuraciones necesarias a realizar para su despliegue.

- `AMAuthIMS` es la clase principal del módulo de autenticación, extendiendo la clase `AMLoginModule` para ser identificada como un módulo de autenticación extensión de OpenSSO. Es la encargada de gestionar las peticiones de autenticación de los diferentes usuarios y gestionar los callbacks enviados en las mismas, para después procesar los resultados obtenidos.

  Los principales atributos que posee esta clase son:

`String fullname`	Variable que almacena el nombre del usuario que ha accedido al método de autenticación.
`String sipuri`	Variable que almacena la IMPU de dicho usuario para poder realizar el intercambio de mensajes SIP con él.
`Pincipal userPrincipal`	Variable que almacena el perfil de autenticación asociado al usuario que se encuentra utilizando el módulo.
`MessageSender ms`	Instancia de la clase que implementa el intercambio de mensajes con el terminal IMS del usuario.

Las principales funciones definidas son:

`void init (Subject subject, Map sharedState, Map options)`	Inicializa el módulo de autenticación y queda a la espera de solicitudes de los usuarios que quieran acceder a él. • subject: Es la entidad asociada al "sujeto" que inicia la autenticación. • sharedState: Representa los estados disponibles para el módulo de autenticación. • options: Representa las diferentes opciones asociadas al módulo de autenticación.

int process(Callback[] callbacks, int state)	Procesa las peticiones de autenticación del usuario, comprobando la validez de las mismas y actualizando la base de datos de peticiones asociadas a los usuarios. • callbacks: Es el array recibido con los requisitos solicitados por el módulo de autenticación en un estado determinado. • state: Estado en el que se encuentra la autenticación y al que van asociados los callbacks anteriores. Devuelve -1 si la autenticación ha sido satisfactoria, 0 si ha habido un error en la misma u otro valor entero dependiendo del estado en que se encuentre el proceso: 1- Autenticación iniciada. 2- Autenticación en proceso. 3- SIP URI no asociada con perfil de usuario. Descubrimiento OpenSSO requerido. 4- Usuario desconocido.
Principal getPrincipal()	Comprueba si existe un perfil de usuario del módulo de autenticación asociado al mismo y en caso de no existir, lo crea. Devuelve el "perfil principal" asociado al usuario.

- `MessageSender` es una clase que implementa SipServlet y que es instanciada por el módulo de autenticación para llevar a cabo el proceso de intercambio de mensajes SIP con el terminal IMS del cliente.

  Sus principales atributos son:

SipFactory sipFactory	Variable que permite la creación de objetos SIP necesarios en la ejecución de aplicaciones, tales como peticiones SIPServlet.

SipSessionsUtil sipSessionsUtil	Permite manejar las sesiones SIP creadas.

Los métodos definidos en la clase se describen a continuación:

String sessionKey (SipServletRequest req)	Devuelve la clave de la sesión establecida, en un String. • req: Es la petición SIP recibida.
void init (ServletConfig servletConfig)	Inicializa el módulo con la configuración del Servlet e intenta obtener los atributos SipFactory y SipSessionsUtil asociados. • servletConfig: Es la configuración del Servlet.
void authenticate(UserRequest ur, String description, String quantity, String modulus, String exponent)	Recibe la petición de autenticación del usuario y genera un mensaje de autenticación con formato AuthMessage y lo envía al terminal IMS del comprador. A su vez, en caso de no haber obtenido los atributos SipFactory y SipSessionsUtil en la inicialización, los obtiene a través de JNDI. • ur: Petición de autenticación del usuario recibida por el módulo de autenticación. • description: Descripción de la compra a validar. • quantity: Cantidad de la transacción a validar. • modulus: Módulo de la clave pública del comprador. • exponent: Exponente de la clave pública del comprador.
boolean checkMessage(SipServletRequest request)	Recibe el mensaje SIP con la respuesta del usuario a través de su terminal IMS, valida la firma del mismo, realiza el parseo a formato AuthMessage y valida el resultado de la confirmación del pago, estableciendo el

	estado de autenticación actual asociado a la petición del usuario. • req: Es la petición SIP recibida del terminal IMS del usuario y remitida por el método de recepción de mensajes. Devuelve el estado de la comprobación del mensaje: “true” si se ha realizado con éxito, y “false” si ha ocurrido algún error.
void doMessage(SipServletRequest request)	Recibe las peticiones SIP del tipo MESSAGE, procede a comprobar su contenido y envía una respuesta de confirmación al remitente. • req: Es la petición SIP recibida. En este caso será la proveniente del terminal SIP del usuario como respuesta a la petición de autenticación y confirmación del pago originado en esta misma clase.

- `AuthMessage` es una clase que representa el mensaje de autenticación intercambiado entre el módulo de autenticación y el terminal IMS del usuario.

  Los principales atributos de dicho mensaje son:

messageTypes msgType	Tipo del mensaje de autenticación, pudiendo adoptar el valor de petición o respuesta.
String token	Almacena el identificador de la transacción realizada.
String itemDescription	Descripción de la transacción realizada.
int quantity	Importe de la transacción realizada.
Document doc	Documento asociado a la transacción realizada.
String approve	Cadena de petición de la aprobación, pudiendo tomar el valor “yes/no” en las peticiones y “yes” o “no” es las respuestas.

Sus principales funciones son las siguientes:

AuthMessage create (String token, String itemDescription, int quantity)	Genera un mensaje de tipo AuthMessage a partir de los valores recibidos como parámetros, estableciendo el tipo de mensaje correspondiente y el valor del atributo approve adecuado en cada caso. Además, una vez generados los elementos asociados a cada uno de los componentes del mensaje, genera el documento "doc" asociado a la compra. • token: Identificador de la transacción. • itemDescription: Descripción de la transacción. • quantity: Importe de la transacción. Devuelve el mensaje de autenticación generado.
AuthMessage parse(String content)	Parsea la cadena recibida como parámetro convirtiéndola a formato mensaje AuthMessage. • content: Contiene el mensaje de autenticación en formato cadena de caracteres. Devuelve el mensaje de autenticación generado.
String getXML()	Genera un documento XML a partir del mensaje de autenticación AuthMessage actual. Devuelve la cadena de caracteres con el documento XML generado.
boolean isApproved()	Comprueba si el pago se ha aceptado o no. Devuelve "true" si el pago se ha aceptado, o "false" en caso contrario.

5.3. Aplicación de pasarela de pago

En el presente apartado se detalla la aplicación de pasarela de pago que permite al usuario realizar el pago de las compras realizadas en un comercio electrónico de forma segura.

El proveedor de servicio de pasarela de pago se encargará de proporcionar al usuario un soporte para pago seguro a través de un terminal móvil IMS. Para ello, hará uso del módulo de autenticación comentado en el apartado anterior.

Una vez seleccionada la compra en el comercio electrónico, el usuario será redirigido a la aplicación de pasarela de pago. Posteriormente, dicha pasarela de pago obtendrá la información del pago a través de la sesión SSO establecida anteriormente por el usuario al identificarse mediante OpenID y se comunicará con el módulo de autenticación para verificar el pago con el terminal IMS del usuario.

5.3.1. Descripción de la aplicación

La aplicación de pasarela de pago implementada proporcionará un canal para el pago seguro en comercios electrónicos. Dicha aplicación se encontrará federada con OpenSSO, para poder tener acceso a la información de identidad asociada a los usuarios identificados en dicho sistema.

En la siguiente figura se detalla la funcionalidad de la aplicación de pasarela de pago dentro del sistema completo desarrollado:

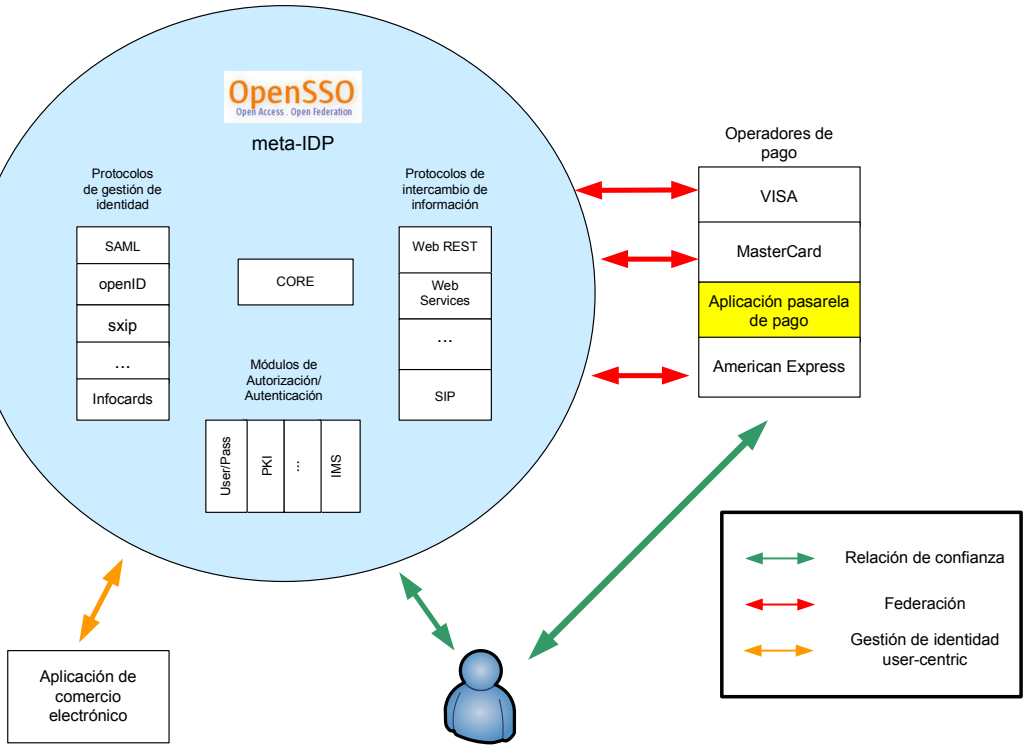

Figura 5.6: Funcionalidad de la aplicación de pasarela de pago en el sistema

Previamente a la descripción del protocolo de funcionamiento de la pasarela de pago, establecemos los requisitos necesarios para que éste pueda iniciarse, que a su vez se corresponden con los mostrados en la Figura 5.7 :

1. Como se indicaba en 5.2, el usuario debe encontrarse registrado en la red IMS a través de su terminal IMS. Además, debe tener activado el servicio de mensajería instantánea.

2. El comercio debe haber redirigido al usuario a la pasarela de manera satisfactoria y haberse identificado mediante OpenID, iniciando así una sesión SSO.

   Una vez cumplidos los dos puntos anteriores, el funcionamiento es el siguiente:

3. La pasarela de pago recupera los datos de la sesión SSO establecida a través de las cookies que viajan en el navegador.

4. Verifica el nivel de autenticación del usuario en el sistema y, al requerirse una mayor confianza para efectuarse el pago, se inicia la autenticación con el módulo descrito anteriormente en 5.2.

5. Una vez lanzado el módulo de autenticación, la pasarela de pago queda a la espera de la confirmación de envío de mensaje de respuesta por el usuario.

6. Dependiendo de si el usuario ha confirmado el envío del mensaje o ha cancelado el proceso, la pasarela de pago remite el nuevo callback al módulo de autenticación.

7. El módulo de autenticación evalúa la transacción y devuelve a la pasarela el estado del proceso de autenticación:

   7.1. Si la autenticación ha sido satisfactoria se redirige al usuario a la página de compra finalizada con éxito alojada en la aplicación del comercio electrónico.

   7.2. Si la autenticación no ha sido satisfactoria se redirige al usuario a la página de error alojada en la aplicación del comercio electrónico.

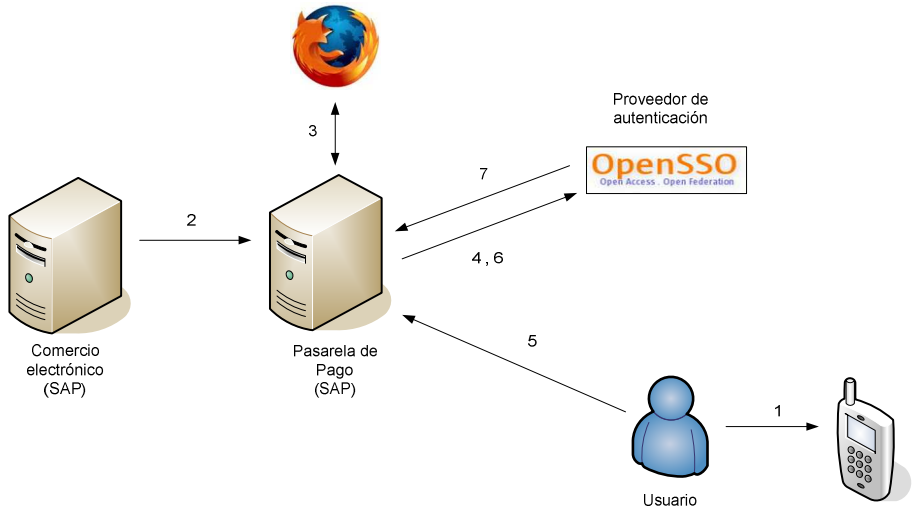

Figura 5.7: Diagrama de funcionamiento de la pasarela de pago

5.3.2. Integración de la aplicación en OpenSSO

Previamente a la descripción del desarrollo realizado para implementar la pasarela de pago, comentaremos brevemente el proceso necesario para integrar una aplicación Web con OpenSSO [40][41][42] y la utilización de tokens para obtener información de las sesiones SSO establecidas con anterioridad.

La siguiente figura muestra el proceso de integración de una aplicación en OpenSSO:

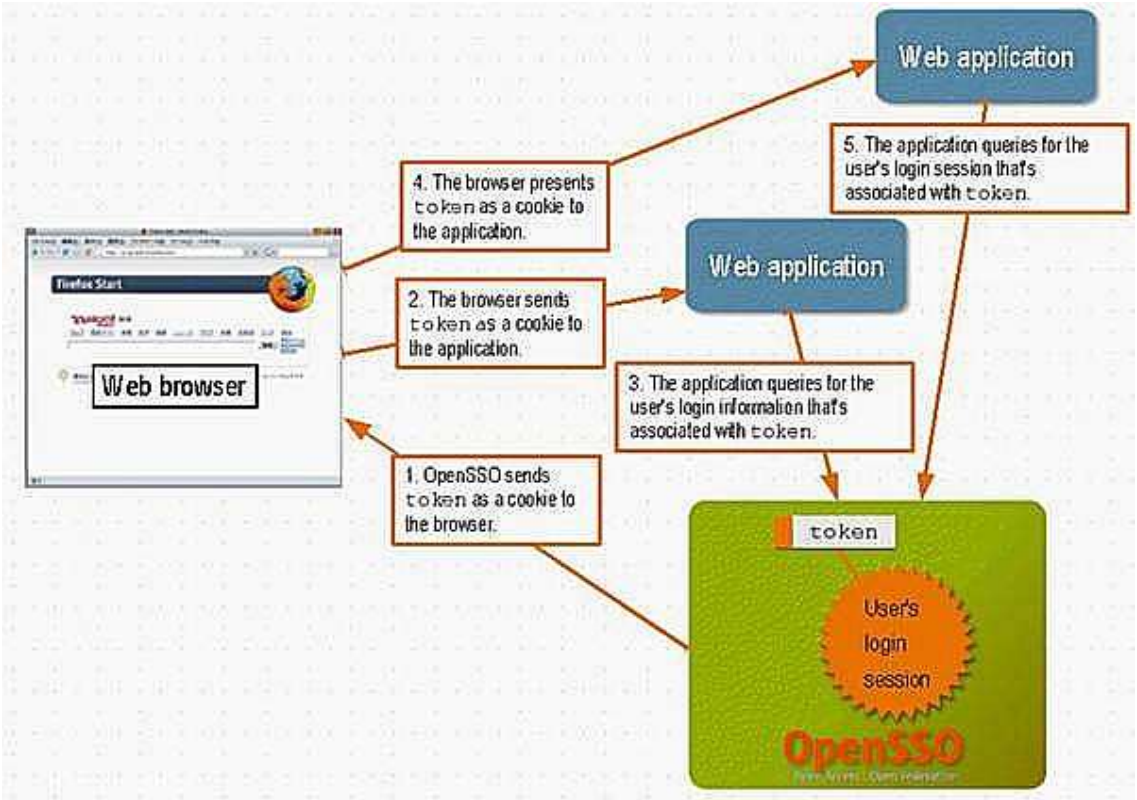

Figura 5.8: Integración de aplicaciones en OpenSSO. Obtenido de [40]

Como muestra la Figura 5.8, una vez el usuario se encuentre registrado en OpenSSO:

1. OpenSSO envía el token al navegador en forma de cookie.

2. El navegador presenta dicha cookie a la aplicación.

3. La aplicación extrae el token y lo procesa para realizar diferentes acciones: solicitar el estado de autenticación del usuario, el nivel de autenticación actual, atributos del perfil de usuario o comenzar un nuevo proceso de autenticación, por ejemplo. Posteriormente, la aplicación envía dicha información de vuelta al navegador.

4. El navegador vuelve a presentar el token en forma de cookie a la aplicación.

5. La aplicación permite acceso al usuario en función de las especificaciones marcadas por OpenSSO y por ella misma.

5.3.3. Implementación de la pasarela de pago

La aplicación J2EE que implementa la pasarela de pago consta de tres entidades que se comunican entre sí para proporcionar el servicio requerido. Como vemos en la Figura 5.9, dicho módulo lo forman dos JSP's [44]: `Index` y `Smswait` y un Servlet HTTP: `GatewayServlet`.

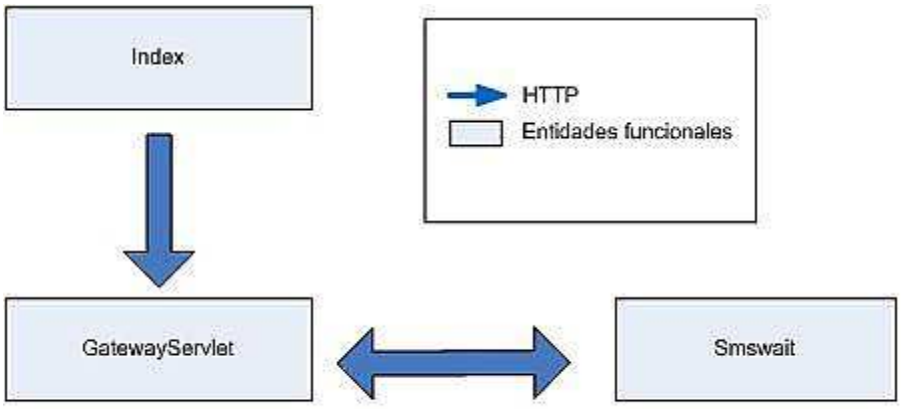

Figura 5.9: Diagrama funcional de la pasarela de pago

- `Index` implementa la página de bienvenida de la pasarela de pago que recibe la petición HTTP proveniente del comercio electrónico, para posteriormente presentar al usuario la oportunidad de aceptar los términos de la transacción a realizar o bien rechazarlos.

  Una vez el usuario seleccione si acepta los términos o no, se le redireccionará al `GatewayServlet` o la página de cancelación de la compra del comercio electrónico, respectivamente.

- `Smswait` implementa la página de espera a confirmación de envío de mensaje de verificación del pago por parte del cliente. Es decir, una vez la pasarela de pago haya enviado al módulo de autenticación la orden de iniciar el proceso de autenticación y éste envíe el mensaje de petición de confirmación al terminal del usuario, dicha pasarela quedará a la espera de que el usuario indique en ésta página si ha enviado ya el mensaje de confirmación o no.

  Para ello se presentan al usuario dos botones:

  - Done: El usuario indica que ya ha enviado el mensaje de confirmación.
  - Cancel: El usuario indica que cancela la transacción o bien que el mensaje enviado carece de validez.

  Una vez pulsado uno de los dos botones, se redirige al usuario de nuevo al `GatewayServlet`.

- `GatewayServlet` es la clase principal de la pasarela de pago, gestionando la interacción del usuario con el módulo de autenticación. Implementa un Servlet SIP, procesando las peticiones provenientes de los JSP's del módulo y validando el estado de la transacción remitido por el módulo de autenticación.

  A continuación se detallan las principales funciones existentes en la clase `GatewayServlet`:

  - **`void processRequest (HttpServletRequest request, HttpServletResponse response)`:** Procesa las peticiones realizadas al Servlet.

    - request, response: Par petición/respuesta asociado al Servlet.

    El funcionamiento del método depende de la procedencia de la petición recibida:

    A) Si la petición proviene de `Index`, se considerará que la pasarela de pago debe iniciar el proceso de autenticación, y se realizarán las siguientes acciones:

    - Obtener los parámetros "Description" y "Quantity" relacionados con la compra realizada a partir de la sesión HTTP actual.
    - Obtener el Token de la sesión OpenSSO existente y validarlo.
    - Obtener los atributos de identidad "Uid" e "IMPU" relacionados con el usuario a partir del token anterior.
    - Comprobar el nivel de autenticación asociado al token.
    - Solicitar requisitos al módulo de autenticación.

- Enviar los requisitos al módulo: “Uid”, “IMPU”, “Description” y “Quantity”.

- Comprobar el estado de autenticación. En caso de encontrase “en progreso”, redirigir al usuario a `Smswait` para esperar su confirmación del envío de mensaje de autorización del pago. En caso de que la autenticación haya fallado, redirigirle a la página de cancelación de la compra del comercio electrónico.

B) Si la petición proviene del botón “Done” de `Smswait`, se considerará que el envío del mensaje se ha realizado correctamente, y se realizarán las siguientes acciones.

- Obtener el contexto de autenticación existente.

- Solicitar requisitos al módulo de autenticación.

- Enviar el requisito al módulo: “Done”.

- Comprobar el estado de autenticación. En caso de autenticación satisfactoria, redirigir al usuario a la página de compra finalizada con éxito del comercio electrónico. En caso de que la autenticación haya fallado, redirigirle a la página de cancelación de la compra del comercio electrónico.

C) Si la petición proviene del botón “Cancel” de `Smswait`, se considerará que la transacción se ha cancelado, y se realizarán las siguientes acciones.

- Enviar el requisito al módulo: “Cancel”.

- Comprobar el estado de autenticación. Al encontrarnos siempre en este caso con que la autenticación ha fallado, se redirige al usuario a la página de cancelación de la compra del comercio electrónico.

o **`void addLoginCallbackMessage(Callback[] callbacks, String uid, String sipuri, String description, String quantity, String modulus, String exponent)`:** Se encarga de almacenar los “callbacks” o requisitos necesarios para iniciar la autenticación en el módulo IMS.

- callbacks: Requisitos del módulo de autenticación para el estado inicial.

- uid: identificador del usuario.

- sipuri: Uri del terminal IMS del usuario.

- description: Descripción de la transacción que está siendo validada.

- quantity: Importe de la transacción que está siendo validada.

- modulus: Módulo de la clave pública del cliente.

- exponent: Exponente de la clave pública del cliente.

o **`void addDoneCallbackMessage(Callback[] callbacks)`:** Se encarga de almacenar los "callbacks" o requisitos del módulo de autenticación cuando se encuentra "en progreso" y se ha pulsado el botón "Done".

- callbacks: Requisitos del módulo de autenticación para el estado "en progreso" tras haber pulsado el botón "Done".

o **`void addCancelCallbackMessage(Callback[] callbacks)`:** Se encarga de almacenar los "callbacks" o requisitos del módulo de autenticación cuando se encuentra "en progreso" y se ha pulsado el botón "Cancel".

- callbacks: Requisitos del módulo de autenticación para el estado "en progreso" tras haber pulsado el botón "Cancel".

5.4. Aplicación cliente

Para el desarrollo del sistema de pago seguro en comercios electrónicos realizado en este proyecto, así como para asistir al usuario a la hora de responder a los mensajes de la pasarela de pago, se requiere el uso de una aplicación cliente que gestione las peticiones de confirmación de pago provenientes de dicho módulo.

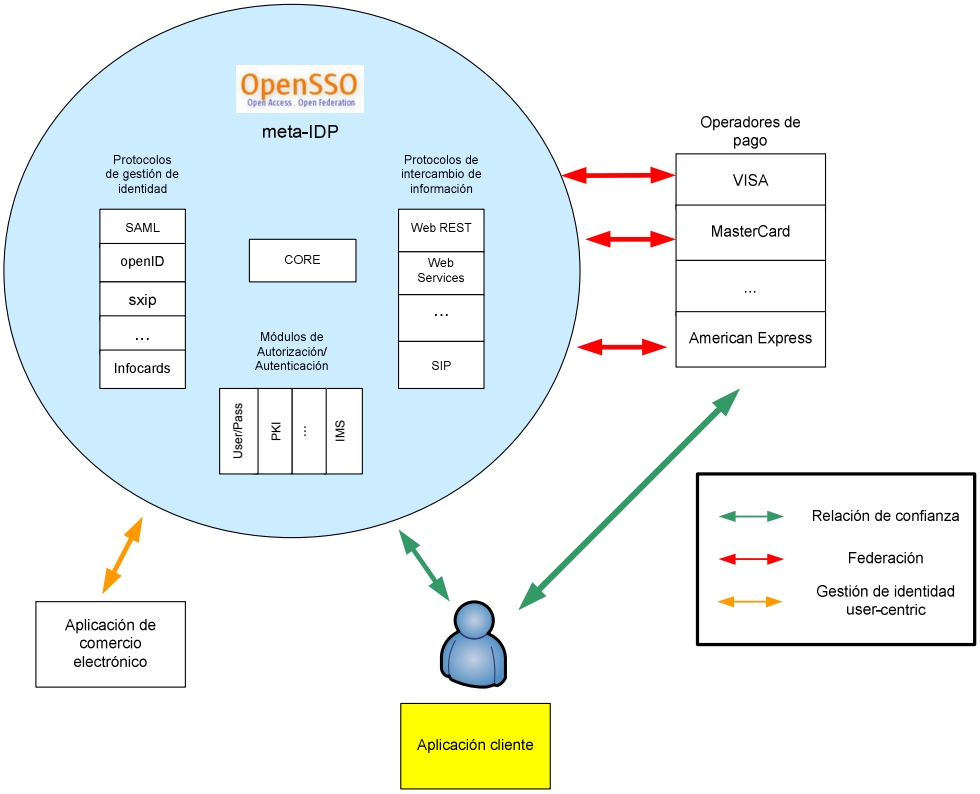

Figura 5.10: Funcionalidad de la aplicación cliente en el sistema

En la actualidad, existen numerosos programas y aplicaciones que realizan la función de cliente en una red IMS, tales como UCT-Client, Monster The Client [45] o Mercuro.

Frente a UCT-Client o Mercuro, Monster The Client proporciona una API con librerías además de una guía de desarrollo, lo que posibilita del desarrollo de plugins o módulos personalizados para añadir al funcionamiento básico de la aplicación.

En nuestro caso, deseamos implementar un módulo con una interfaz gráfica vistosa e intuitiva que permita al usuario verificar transacciones mediante la pulsación de botones de su terminal IMS. Por ello, "Monster The Client" [45] ha sido la aplicación cliente seleccionada para nuestro desarrollo.

Mediante el uso de librerías proporcionadas por sus creadores, el desarrollo de un nuevo módulo es sencillo e intuitivo, como se mostrará más adelante en éste capítulo. No obstante en primera instancia se especificarán los requisitos del módulo a diseñar y la funcionalidad que deberá presentar.

5.4.1. Descripción de la aplicación cliente

Previamente al uso del servicio de mensajería, deben cumplirse los siguientes requisitos:

- El usuario debe haber conectado el terminal y haberse registrado a través de él a la red IMS con la IMPU que tenga activada en su perfil OpenSSO.

  Como se indica en el apéndice B.3 de este texto, el usuario deberá configurar su terminal IMS, proporcionando información acerca de la red IMS a la que se conectará, así como los identificadores que tendrá asignados en dicha red.

  La provisión de dichos parámetros es fundamental, debido a que son necesarios para que el módulo de autenticación localice al terminal IMS dentro de la dicha red.

- El usuario debe haberse creado un perfil en el servicio de mensajería.

  Como se indica en el apéndice B.3 de este texto, será necesario registrarse en el servicio de mensajería para poder identificar al usuario. Si el usuario no se encuentra registrado en dicho servicio no podrá hacer uso de dicha aplicación.

- El usuario debe activar el servicio de mensajería previamente a aceptar los términos de la compra en la pasarela de pago.

  Como se indica en el apéndice B.3 de este texto, el usuario deberá iniciar la aplicación cliente y pulsar el botón que activa el módulo de mensajería instantánea, para que quede a la escucha de mensajes de confirmación de pago provenientes del módulo de autenticación.

En la figura 5.11 se especifica el funcionamiento básico del servicio de mensajería instantánea instalada en el terminal IMS. A continuación se detallan los correspondientes pasos:

1. La aplicación espera a la recepción de mensajes de confirmación de compra de la pasarela de pago.

2. Una vez recibido el mensaje, presenta al usuario la opción de aceptar la transacción o denegar el pago.

3. Se crea un mensaje de respuesta firmado digitalmente mediante algoritmo RSA y se envía de vuelta al módulo de autenticación.

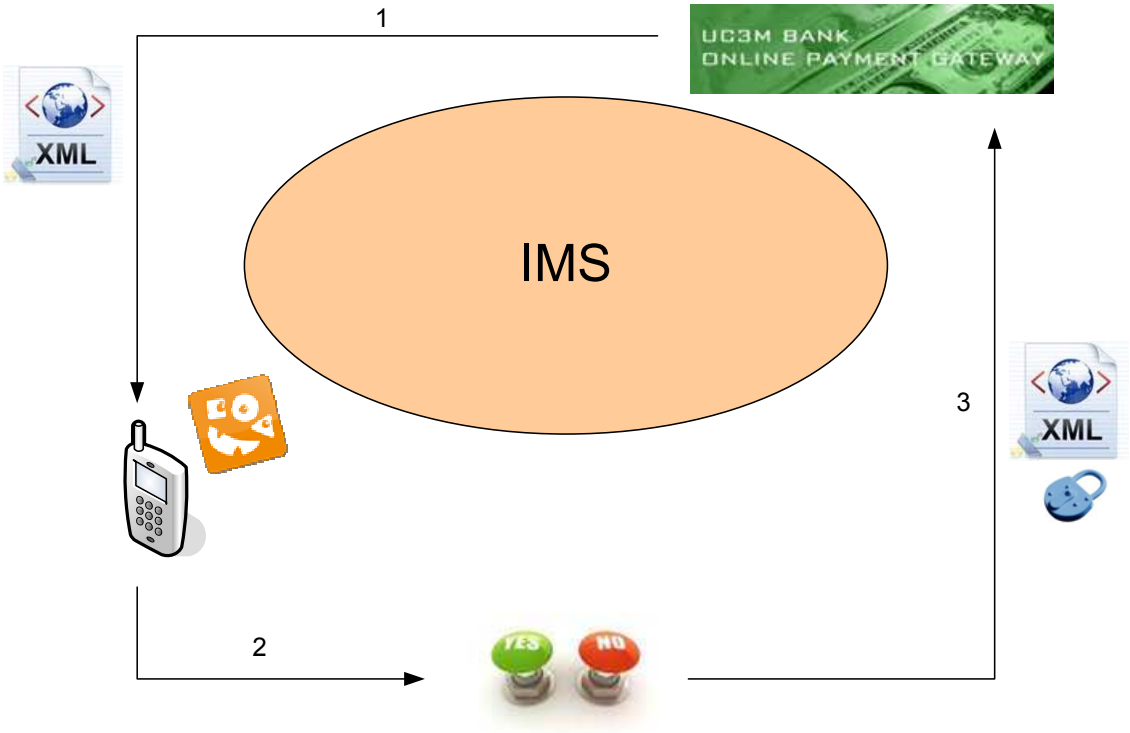

Figura 5.11: Diagrama de funcionamiento de la aplicación cliente

5.4.2. Desarrollo del plugin de mensajería instantánea

Debido a la gran cantidad de clases implementadas en el desarrollo del plugin de mensajería, tan sólo se describen a continuación las principales entidades cuya funcionalidad merece ser comentada:

- `Module`**:** Clase que representa al módulo y sirve de punto de entrada para el plugin desarrollado dentro del marco de trabajo.

  Así mismo, define las dependencias existentes del módulo en cuestión con el resto de módulos existentes.

  Sus principales funciones son:

void load()	Se ejecuta cuando se inicia el módulo. Accede al configurador de preferencias del sistema y lo configura para que el módulo creado sea accesible desde él, registrando su configuración de apariencia. Una vez configurado lanza el módulo.
void unload()	Se ejecuta cuando se desconecta el módulo, desconfigurando el mismo del sistema.
String getName()	Devuelve el nombre del módulo.

- `ModuleWorkItem:` Clase que define la lógica básica del plugin desarrollado, estableciendo su modo de funcionamiento y registrando los servicios que dicho plugin ofrecerá al usuario.

  Configura su representación dentro de la aplicación y la manera en la que será activado y desactivado.

- `SMSProfile:` Define la configuración de los perfiles de usuario creados en el plugin definido, estableciendo el nombre de usuario como único atributo de dicho perfil.

- `SMSViewImpl:` Define la interfaz gráfica que presentará el plugin de servicio de mensajería automática al usuario.

  En el caso particular de este proyecto, el plugin mostrará un mensaje de bienvenida y un botón para obtener el perfil del usuario que se encuentra asociado a dicho módulo.

- `SMSWorkItem:` Define el funcionamiento del plugin implementado, extendiendo a la interfaz `MessageListener`, lo que permitirá al plugin reaccionar ante eventos relacionados con la recepción o envío de mensajes.

  Sus principales funciones y métodos son:

void onRunStarted()	Carga la configuración inicial del plugin, mostrando el mensaje de bienvenida y quedando a la espera de la recepción de un mensaje.
void serviceInject(MessageService messageService, MessagingService messagingService)	Inyecta los servicios de mensajería para la manipulación de mensajes SIP y de información mediante ventanas emergentes, respectivamente.

void setProfile(SMSProfile profile)	Establece el perfil de usuario recibido.
void messageReceived(MessagingService arg0, Message arg1)	Se ejecuta ante la llegada de un nuevo mensaje a la aplicación. En primer lugar muestra el mensaje recibido en una ventana emergente, para posteriormente solicitar al usuario la confirmación del pago mostrado anteriormente. Dependiendo de la elección del usuario se creará un documento XML con la validación/denegación de la transacción firmado digitalmente y será enviado de vuelta al módulo de autenticación.
void messageSendFailed(MessagingService arg0, Message arg1, String arg2)	Se ejecuta en caso de producirse un error al enviarse el mensaje, mostrando un mensaje en una ventana emergente.
void messageSent(MessagingService arg0, Message arg1, String arg2)	Se ejecuta al enviarse un mensaje, mostrando que se ha enviado correctamente en una ventana emergente.

5.5. Conclusiones

El módulo de verificación de pagos desplegado provee al sistema de una gran robustez frente a fraudes electrónicos. Mediante la pasarela de pago, protege la información referida a la transacción económica entre el comercio electrónico y el usuario, al intercambiar dicha información cifrada y a través de una red segura como es IMS.

A su vez, el uso de dos canales de comunicación diferenciados (Internet y red de telefonía móvil) potencia la robustez del sistema diseñado, ya que es mucho más difícil comprometer varias redes de manera simultánea que cada una de ellas por separado.

La provisión de dicho mecanismo de pago seguro estará orientada a potenciar el comercio electrónico. Con ello se pretende motivar tanto a usuarios como a comerciantes a utilizar un nuevo método de pago en un modelo de red convergente, eliminando las desconfianzas existentes en los mecanismos de pago a través de la red actuales.

Por otro lado, se ha diseñado una aplicación cliente que facilita la verificación de transacciones a los usuarios, lo que favorece a la disminución del tiempo empleado en la

realización de compras electrónicas, así como proporciona una mayor comodidad al usuario para verificar dichas compras.

Capítulo 6

Pruebas

El siguiente apartado mostrará las pruebas realizadas sobre el sistema y cada uno de los módulos que lo componen.

A la hora de diseñar el plan de pruebas se observó que la comprobación por separado de cada uno de los módulos que componen el sistema no tendría mucho sentido, ya que el objetivo de las pruebas realizadas en este apartado no es otro que verificar el proceso completo de pago seguro en un comercio electrónico, realizando la verificación del pago a través de un terminal IMS.

Por ello, se ha diseñado una prueba de conjunto en la que se simula un proceso de compra completo en el comercio electrónico. Con ello, se pretende comprobar el buen funcionamiento de los siguientes módulos, así como la interacción entre ellos:

- Sistema de gestión de identidad: Se identificará al usuario mediante OpenID y se proporcionarán al comercio electrónico los datos de identidad de dicho usuario que requiera.

  Para ello se pedirá al usuario que seleccione un producto del comercio electrónico y que posteriormente se identifique mediante su identificador OpenID. Se le redirigirá a su proveedor de identidad OpenID, donde deberá autenticarse y autorizar la provisión de la información de identidad al comercio electrónico.

- Módulo de verificación de pago a través de IMS: Se comprobará que el comercio electrónico redirige correctamente al usuario a la pasarela de pago. Posteriormente el usuario aceptará los términos de la transacción y se iniciará el proceso de verificación.

  Se comprobará así el funcionamiento del módulo de autenticación y de la aplicación cliente, que recibirá el mensaje de verificación del pago proveniente de dicho módulo de autenticación y presentará al usuario el interfaz para aprobar la transacción.

  Tras ser aceptada la compra, el módulo de autenticación comprobará la validez del mensaje de respuesta, así como verificará la identidad del usuario que ha firmado dicho mensaje, evitando así ataques de suplantación de identidad.

  Una vez la transacción haya sido finalizada, la pasarela redirigirá al usuario al comercio electrónico presentándole el resultado de dicha transacción.

Posteriormente a la prueba de conjunto especificada, se planteará una batería de pruebas adicionales, orientada a detectar las vulnerabilidades del sistema y su robustez ante ataques externos de diferente tipo: suplantación, repetición, intercepción…

6.1. Ejecución de un proceso de compra completo

Previamente a la realización de la compra en el comercio electrónico, como se comentó en 3.2, es necesario haber registrado el terminal IMS del usuario a dicha red. Por ello, a continuación se indican los principales pasos necesarios para su conexión a la red y la configuración y puesta en marcha del plugin de mensajería instalado en la aplicación cliente.

En primer lugar, abrimos la aplicación cliente "Monster The Client".

Figura 6.1: Selección de perfil en "Monster the client"

Seleccionamos el perfil creado en la aplicación y nos conectamos a la red IMS (debiendo estar configurado correctamente dicho perfil).

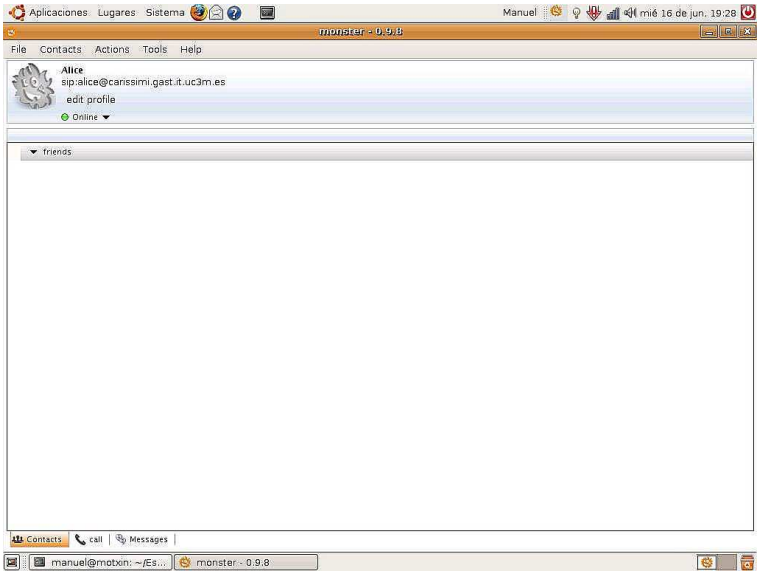

Figura 6.2: Página principal del perfil de usuario en "Monster the client"

A continuación, entramos en Tools -> Preferences, para configurar el plugin diseñado para la aplicación cliente.

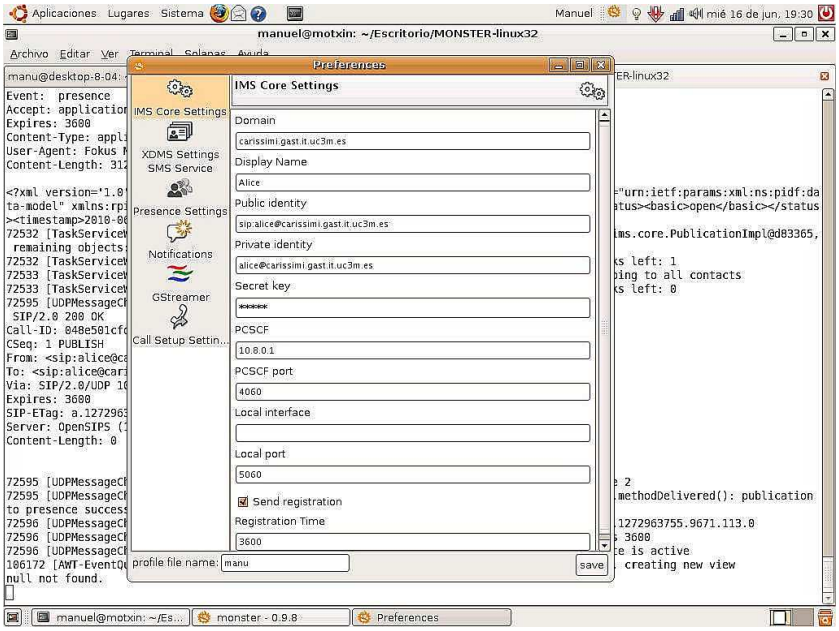

Figura 6.3: Menú de configuración de "Monster the client"

Como podemos observar en la Figura 6.4, en el menú de la izquierda aparece nuestro módulo "SMS Service", que seleccionamos.

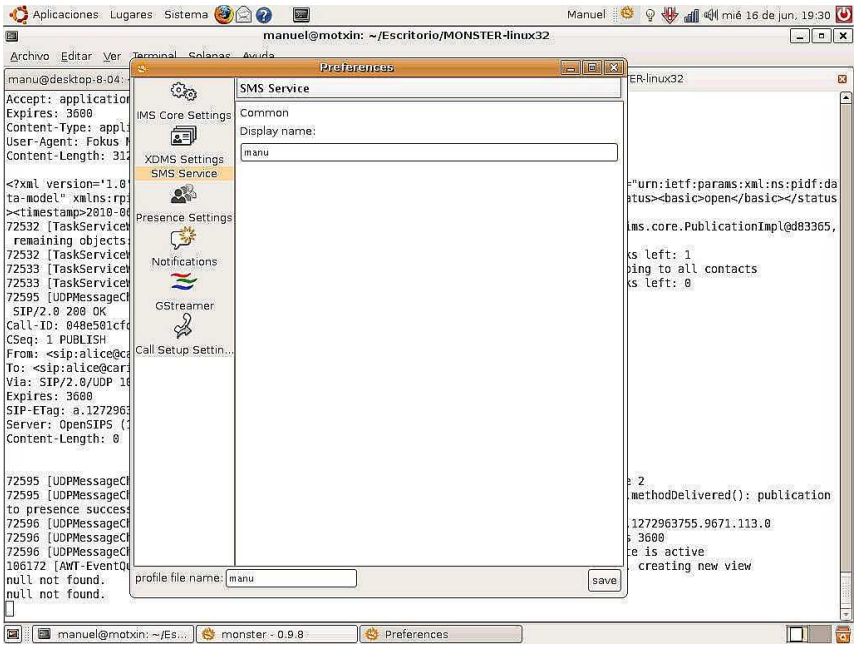

Figura 6.4: Configuración del módulo de mensajería instantánea

Dentro de la configuración del módulo de mensajería, seleccionamos el nombre que daremos a nuestro perfil y nuestra identidad pública para dicho módulo. Una vez actualizados los cambios, pulsamos en "Save" para guardar el perfil de usuario creado.

El plugin se encontrará activo a partir de este momento y quedará a la espera de la recepción de mensajes provenientes del módulo de autenticación.

Para iniciar el proceso de compra, el usuario accede a la página Web del comercio electrónico. El interfaz del comercio electrónico se mostrará como muestra la Figura 6.5:

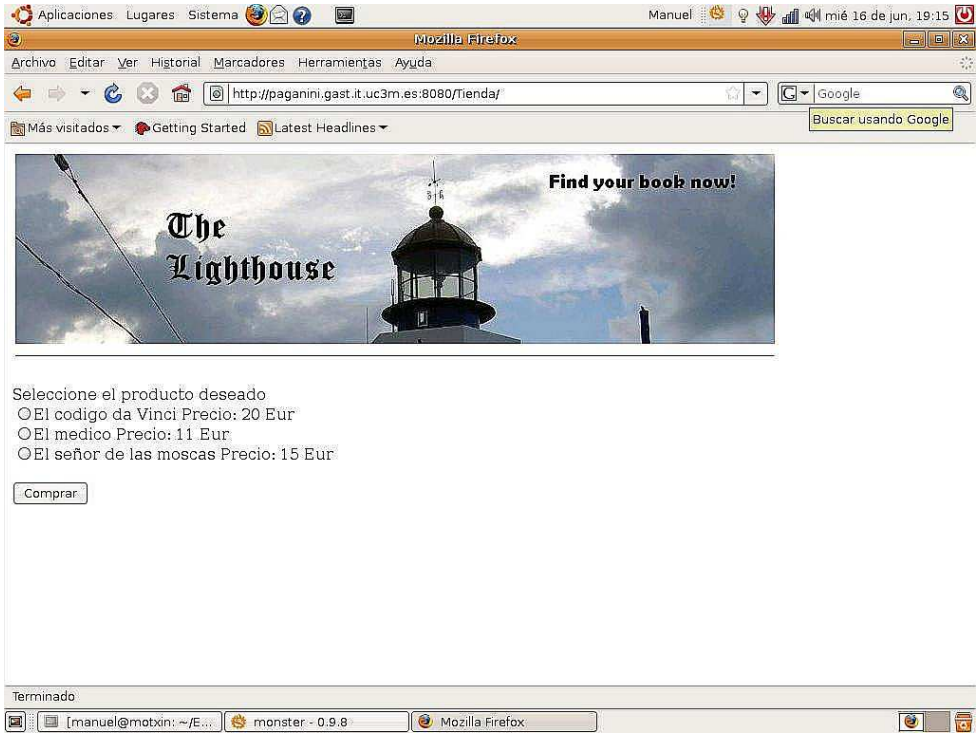

Figura 6.5: Página de bienvenida del comercio electrónico

El comercio electrónico desarrollado se ha orientado a proporcionar de un caso de uso de servicio sobre red IMS. Por ello, el diseño del interfaz de dicho comercio y la complejidad de su funcionamiento no han sido considerados los puntos de mayor interés del sistema.

El comercio electrónico simplemente permitirá al usuario seleccionar un producto de una lista y proceder a su compra, como se mostrará a continuación.

Se observa que la página de bienvenida al usuario al comercio es sencilla e intuitiva, pidiendo al usuario que seleccione el libro que desea comprar y después pulse el botón “Comprar”, como se indica en la Figura 6.6.

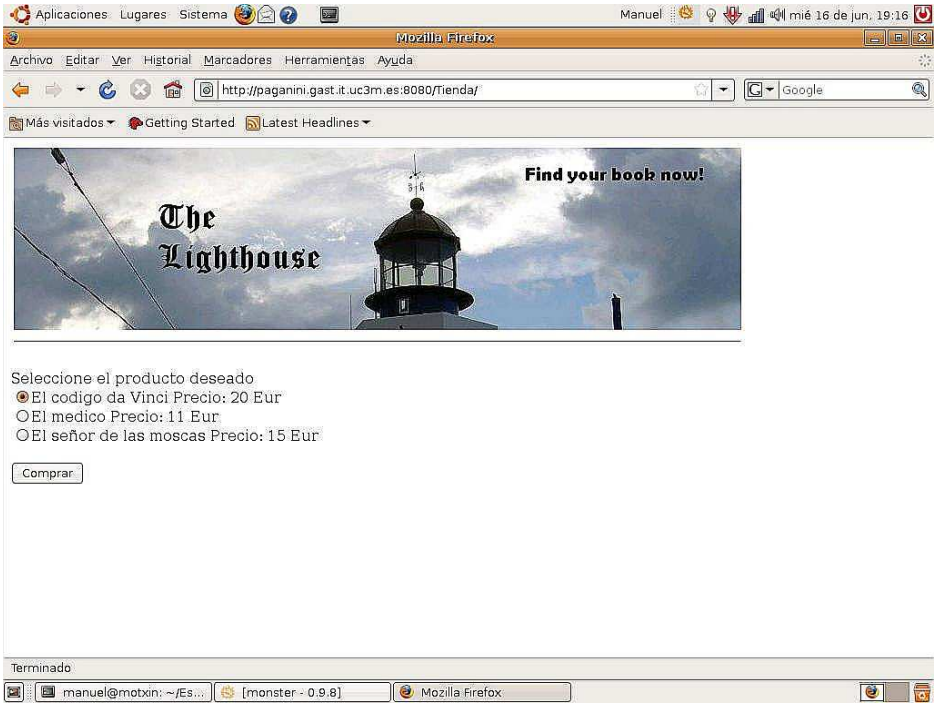

Figura 6.6: Selección de libro

Una vez autorizada la compra, el comercio electrónico nos indicará que es necesario encontrarse registrado en el mismo para poder efectuar la compra deseada. Para ello, nos proporciona la oportunidad de identificarnos mediante nombre de usuario y contraseña (habiéndose producido el registro en el comercio anteriormente) o mediante el uso de OpenID.

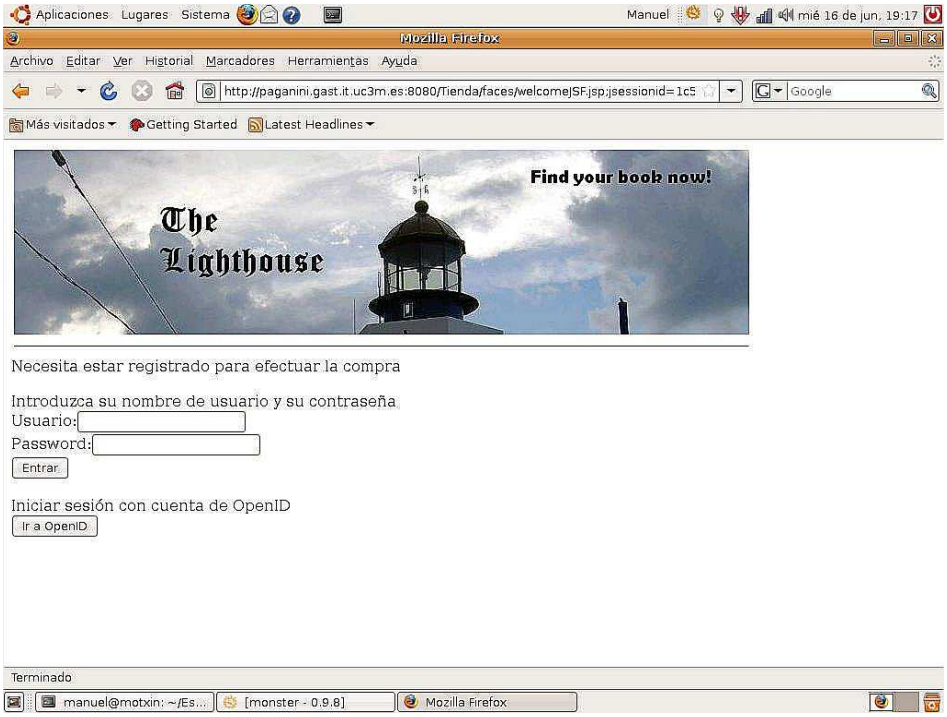

Figura 6.7: Página de identificación del comercio electrónico

La opción de identificación mediante nombre de usuario y contraseña se muestra a modo de que el comercio electrónico ofrezca una perspectiva lo más realista posible, ya que la mayoría de los comercios electrónicos permiten el uso de dicho mecanismo de identificación.

Sin embargo, nos centramos en la identificación mediante OpenID, por lo que la siguiente acción a realizar por el usuario sería la selección de dicha opción mediante la pulsación del botón “Ir a OpenID”.

Seguidamente, el comercio electrónico mostrará la página de registro correspondiente, pidiendo al usuario la introducción de su identificador OpenID y la lista de atributos que necesita obtener de su perfil:

- **Nickname:** Nombre de pila del usuario, para poder activar la sesión de compra y asociar las diferentes peticiones realizadas por dicho usuario.
- **Sip Uri:** Uri asociada al terminal IMS del usuario, para poder autorizar el pago de la compra realizada.
- **Payment Resources:** Pasarela de pago asociada al usuario, para poder redirigirle a ella y que la compra se valide satisfactoriamente.

Como ya se ha comentado anteriormente, la autenticación mediante OpenID permite al usuario seleccionar la información personal a proporcionar el comercio, como vemos en la Figura 6.8.

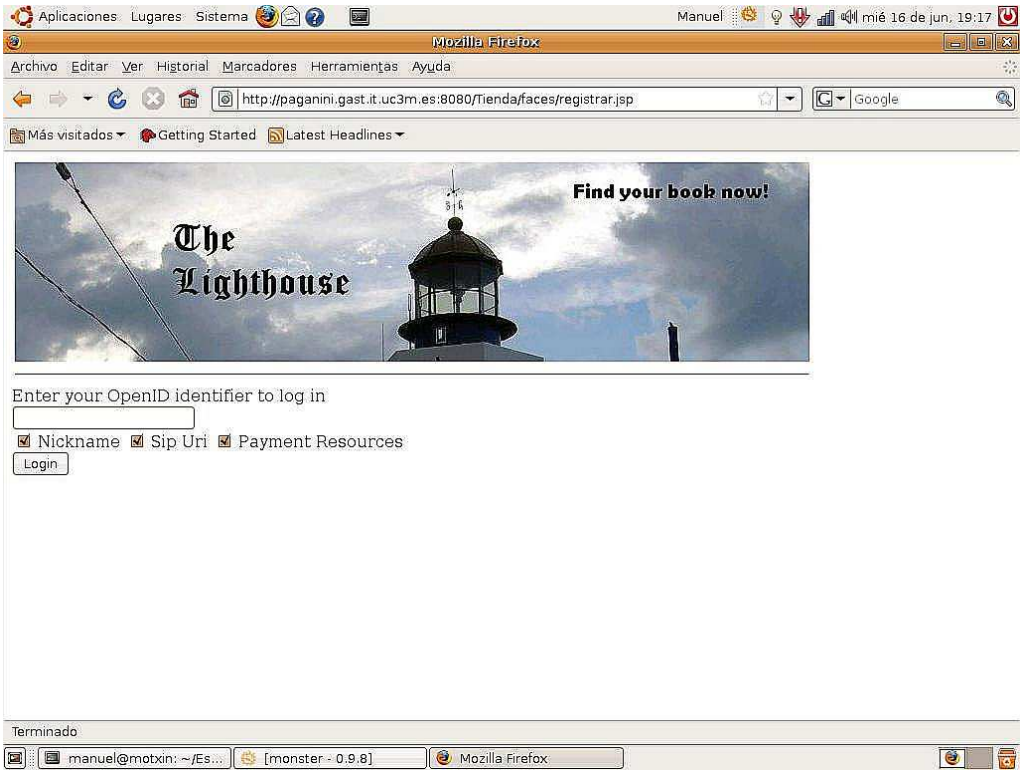

Figura 6.8: Página de identificación mediante OpenID del comercio electrónico

El identificador OpenID de un usuario podrá ser una XRI o una URL, como se describió en 2.3.6.

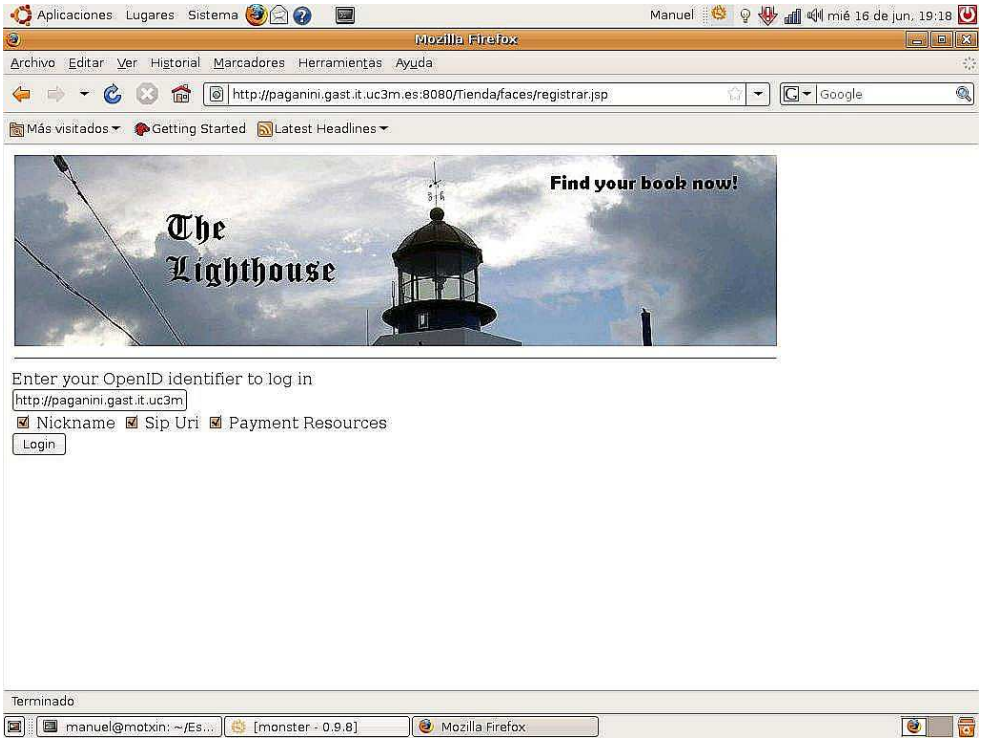

Figura 6.9: Provisión de identificador OpenID al comercio electrónico

El usuario, al pulsar el botón de “Login” será redirigido a su proveedor de identidad OpenID.

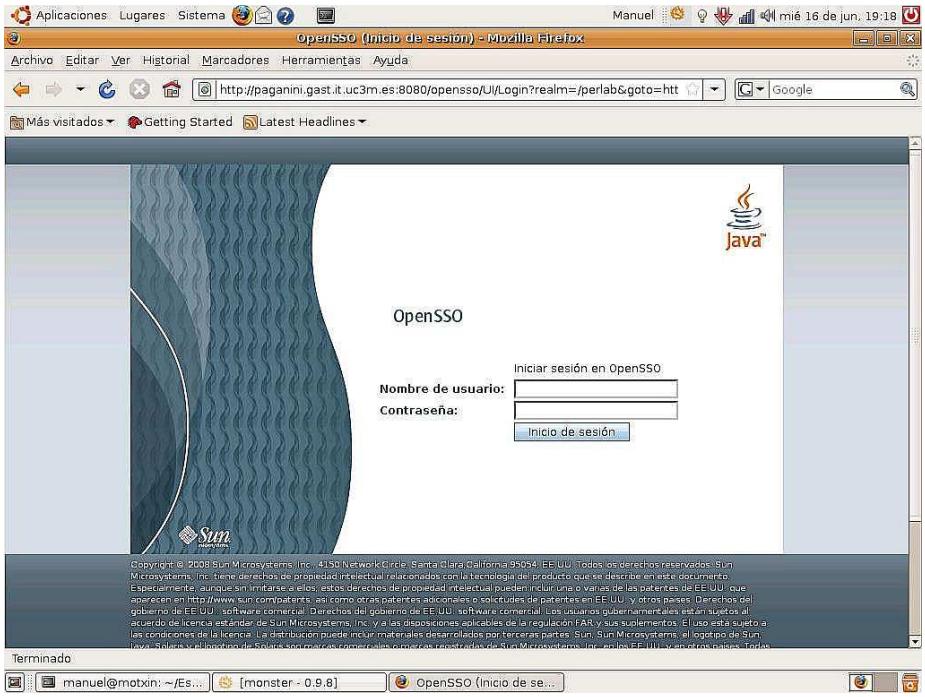

Figura 6.10: Página principal del proveedor de identidad

El proveedor de identidad solicitará al usuario que se autentifique mediante nombre de usuario y contraseña. Una vez introducidos, pulsa “Inicio de sesión”.

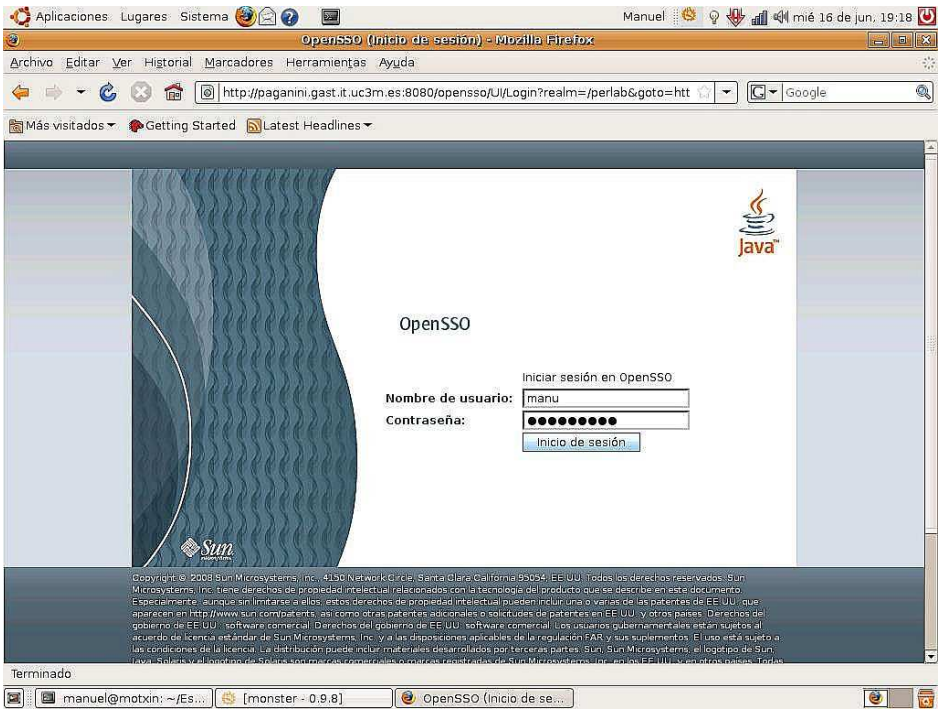

Figura 6.11: Identificación en el proveedor de identidad

El proveedor de identidad mostrará al usuario un interfaz donde muestra la petición de atributos realizada por el comercio.

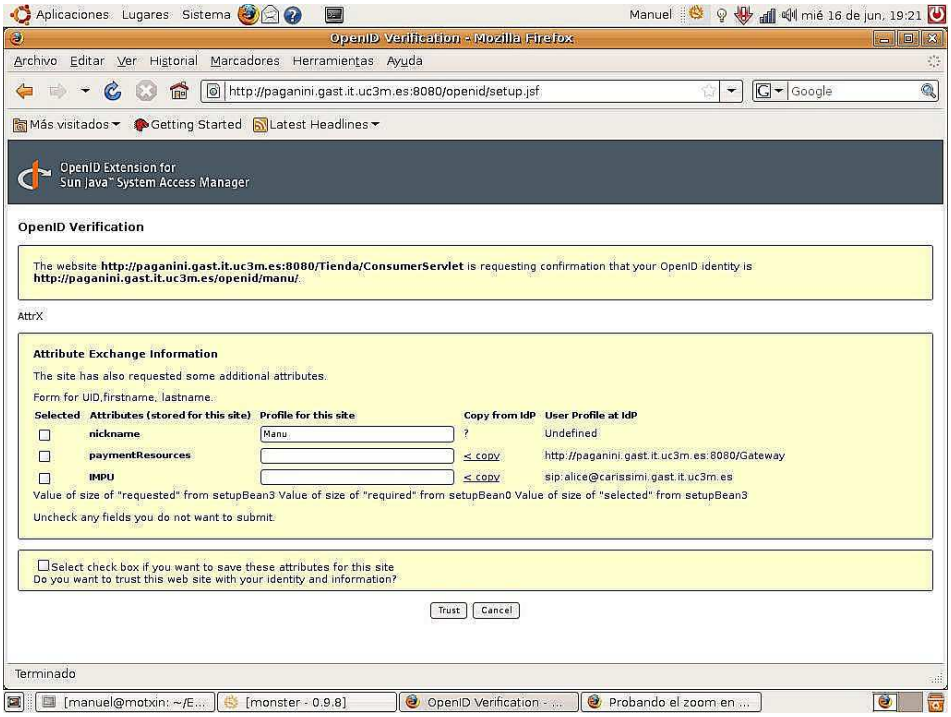

Figura 6.12: Página de verificación del proveedor OpenID

Como vemos en las Figuras 6.12 y 6.13, el cliente podrá proporcionar al comercio electrónico los valores de los atributos establecidos en su perfil de usuario, si bien podrá a su vez introducir nuevos valores introduciéndoles en los pertinentes cuadros de texto. El usuario debe seleccionar a su vez en los cuadros a mano izquierda los atributos que autorice proporcionar a la tienda.

A su vez, el interfaz proporciona al usuario la capacidad de almacenar los atributos proporcionados a un sitio Web determinado seleccionando la opción mostrada en la parte inferior de la página.

Finalmente, mediante la pulsación del botón “Trust”, indicamos al proveedor de identidad nuestra voluntad de proporcionar nuestra información a la página del comercio electrónico. En caso de no tener confianza en dicha página, pulsando “Cancel”, denegaríamos la provisión de los atributos al comercio y la autenticación no se produciría con éxito.

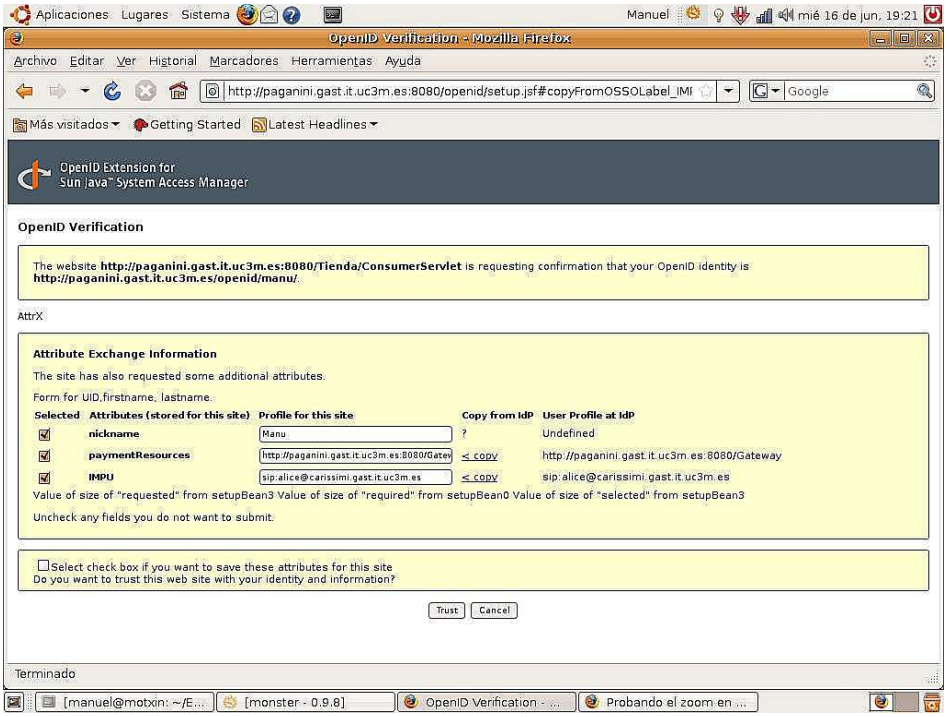

Figura 6.13: Provisión de atributos solicitados por el comercio electrónico

Tras recibir la información del perfil del usuario proveniente del IdP, el comercio examina el atributo Payment Resources para determinar la dirección de la Pasarela de pago asociada al mismo y, posteriormente, redirigirle a ella.

La pasarela de pago, una vez recibida la petición del comercio electrónico, preguntará al usuario si acepta los términos e la transacción correspondiente.

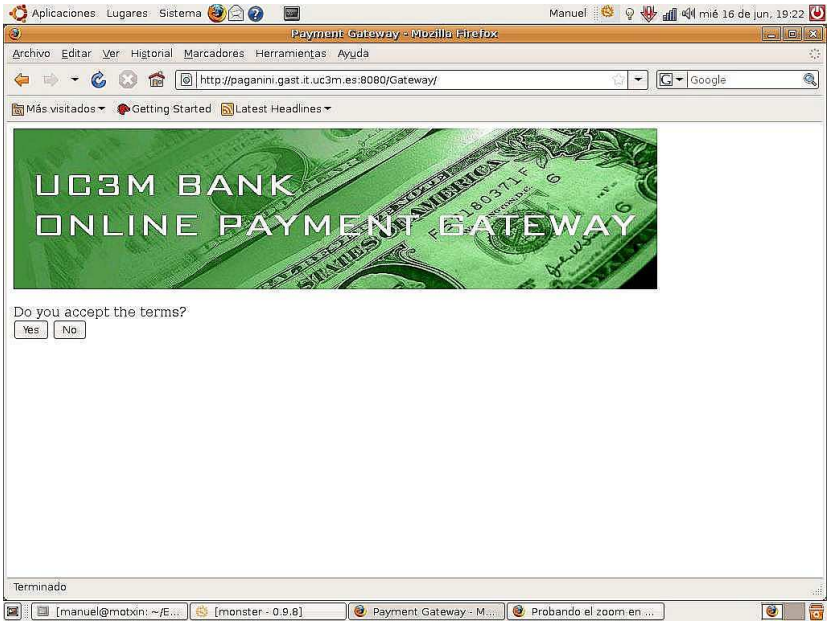

Figura 6.14: Aceptación de términos en la pasarela de pago

Al pulsar sobre el botón "Yes", la pasarela se comunicará con el módulo de autenticación IMS y generará el mensaje de petición de confirmación para mandárselo al usuario. A su vez, la pasarela quedará a la espera de la confirmación de envío de mensaje de vuelta al módulo de confirmación por parte del usuario, como puede observarse en la Figura 6.15.

En caso de no aceptar los términos de compra, el usuario será redirigido a la página de cancelación de compra del comercio electrónico, la cual se comentará más adelante.

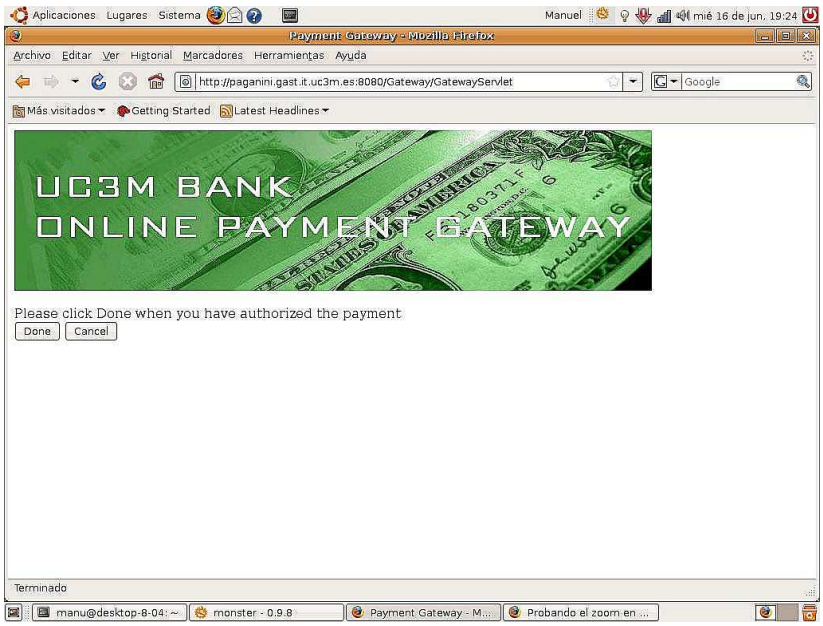

Figura 6.15: Página de aceptación de términos en la pasarela de pago

Ahora, nuestra aplicación cliente, correctamente configurada como se indicó al inicio del presente capítulo, se encargará de recibir el mensaje proveniente del módulo de autenticación y presentar al usuario la oportunidad de confirmar el pago.

Si desplegamos la aplicación cliente en la barra de tareas, podremos ver que la interfaz de la misma muestra que se encuentra ejecutando el plugin de mensajería instantánea y se encuentra a la espera de un mensaje.

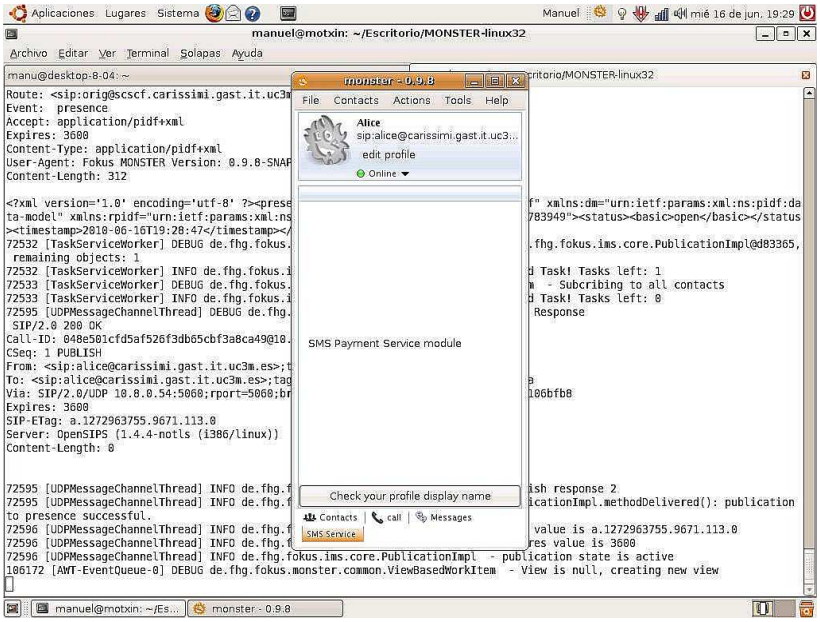

Figura 6.16: Aplicación cliente a la espera de recepción de mensaje

Una vez reciba el mensaje, la aplicación mostrará por pantalla una ventana de información con el contenido del mensaje recibido, para que el usuario pueda comprobar de forma rápida y sencilla las condiciones del pago solicitado.

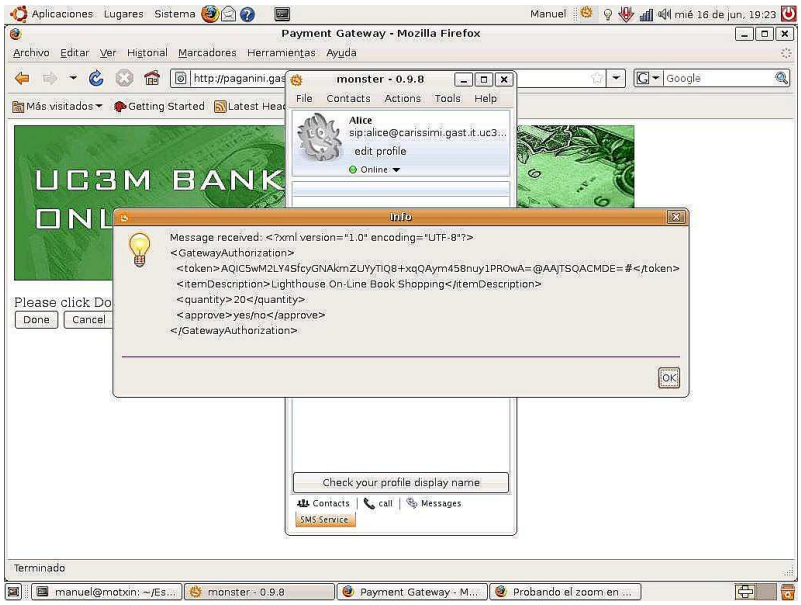

Figura 6.17: Recepción de mensaje en la aplicación cliente

Una vez pulsado “OK”, se presenta al usuario la posibilidad de aceptar el pago o no. En ambos casos, la aplicación generará un mensaje firmado digitalmente con la decisión tomada por el usuario, para después enviarla de vuelta al módulo.

Aceptada o denegada la transacción, el usuario queda a la espera del mensaje, que será notificado mediante una nueva ventana emergente:

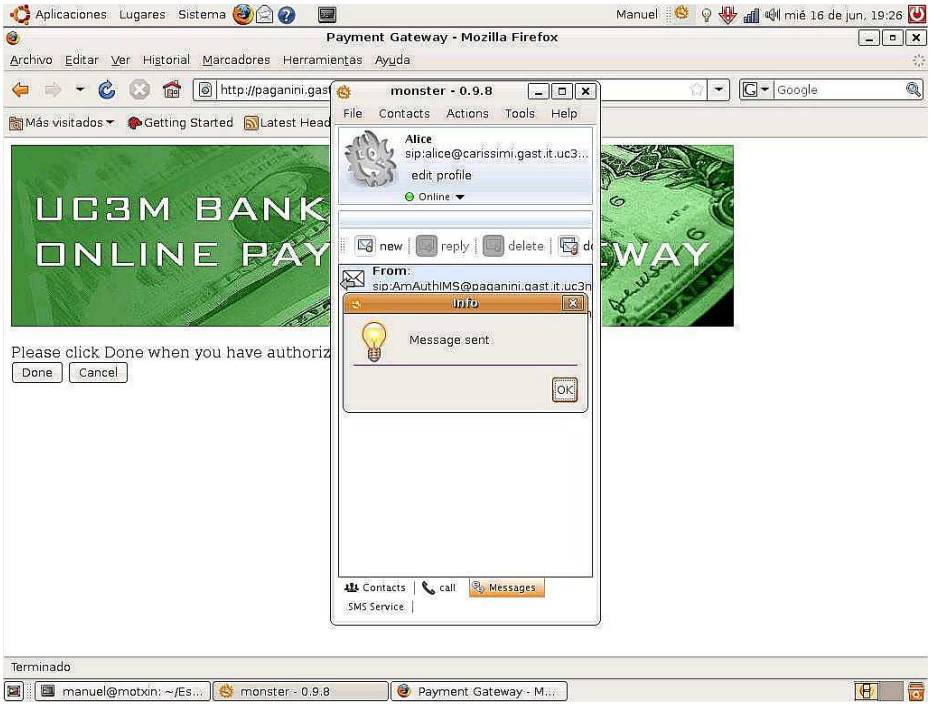

Figura 6.18: Envío de mensaje de confirmación al módulo de autenticación

El siguiente paso será notificar a la pasarela de pago que se ha realizado la confirmación/denegación del pago vía SMS. Para ello, volveremos a desplegar la Web de la pasarela y pulsaremos el botón “Done” en caso de que queramos dar por realizado el proceso, o “Cancel” si queremos cancelar el proceso de compra a pesar de haber confirmado el pago con el terminal IMS.

Posteriormente, la pasarela de pago interactuará con el módulo de autenticación IMS para determinar si la autenticación ha sido satisfactoria o no y si el pago ha sido finalmente validado o cancelado por el usuario.

En caso de que la transacción se haya finalizado correctamente, la pasarela de pago redirigirá al cliente a la página de finalización de compra del comercio electrónico, como muestra la Figura 6.19. En caso contrario, es decir, que se haya producido algún error en el proceso de compra, se redirigirá al usuario a la página de error de compra del comercio electrónico (Figura 6.20).

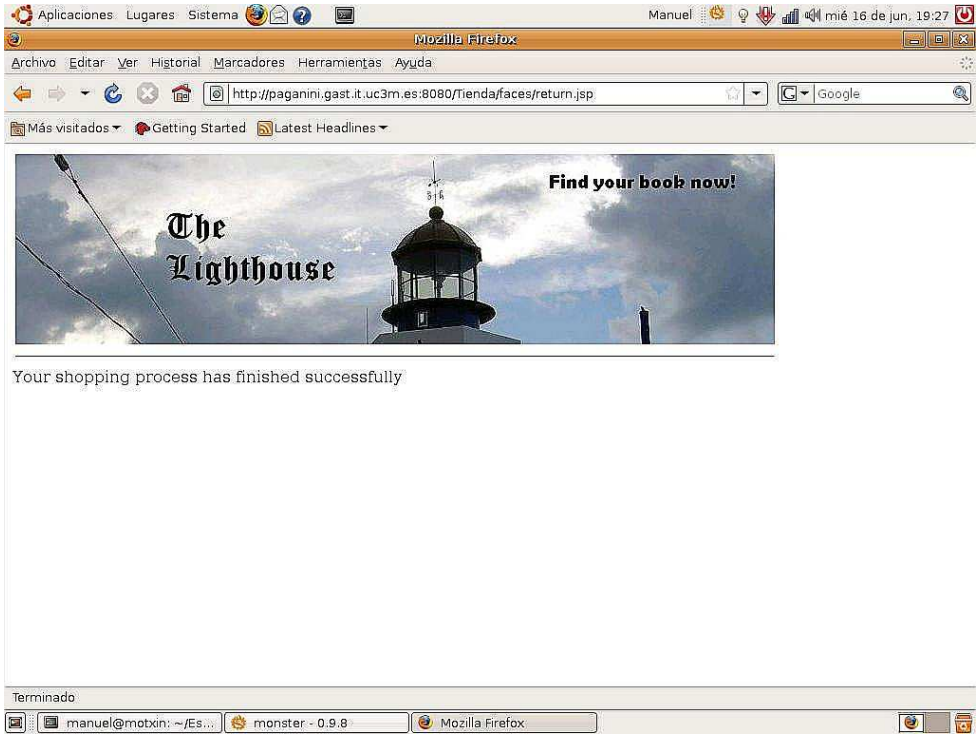

Figura 6.19: Página de finalización de compra satisfactoria

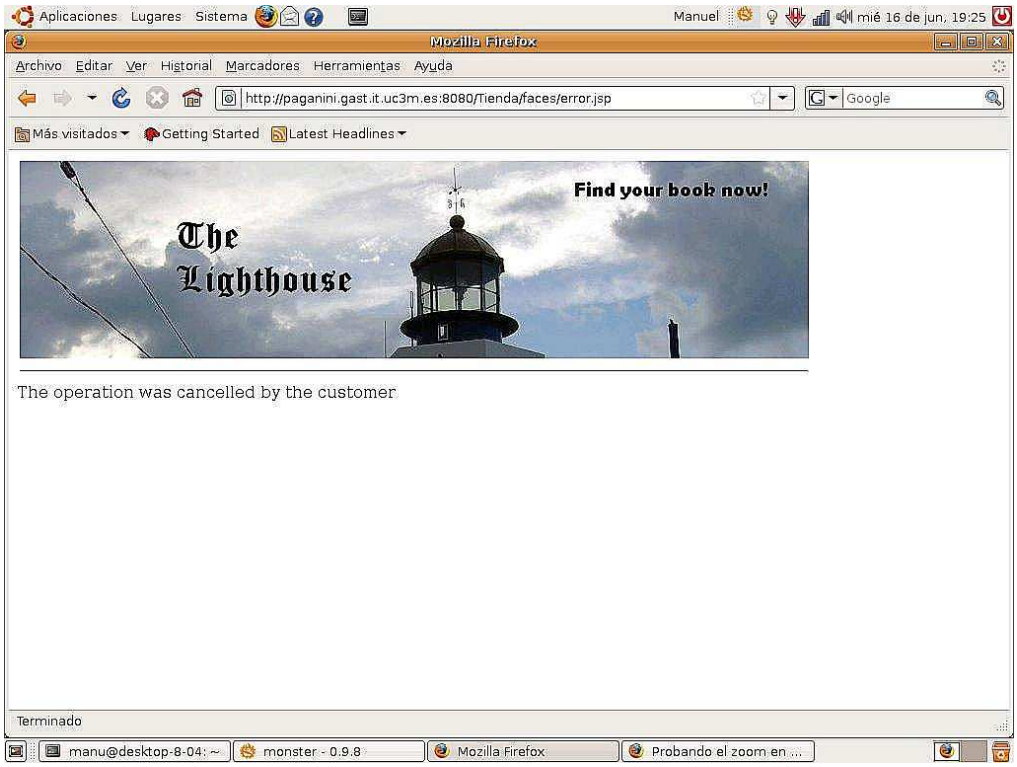

Figura 6.20: Página de error en la compra

6.2. Pruebas de seguridad

En la siguiente tabla se muestran las diferentes pruebas de seguridad realizadas al módulo para comprobar robustez frente a ataques de diferente origen:

Prueba realizada	Resultado	Conclusiones
Pulsar el botón de “Done” antes de enviar el mensaje de confirmación desde la aplicación cliente.	La pasarela de pago no acepta dicho mensaje y el proceso de autenticación queda en progreso.	El funcionamiento es bueno aunque podría implementarse la posibilidad de volver a mostrar la pantalla de Done/Cancel en caso de que no se haya mandado el mensaje desde el cliente.
Enviar el mensaje de confirmación desde otro terminal IMS.	La transacción se cancela debido a que la firma no se verifica correctamente.	Se cumple la protección frente a ataques de suplantación de identidad.
Esperar varios minutos hasta pulsar el botón “Done” una vez enviado el mensaje de confirmación desde la aplicación cliente.	La pasarela de pago no acepta dicho mensaje y el proceso de autenticación queda en progreso.	La sesión expira con el paso de varios minutos y este hecho, a pesar de que requiere de una rápida respuesta por parte del usuario, protege a la transacción de ataques de intercepción, al ser un proceso rápido y dinámico.
Enviar varios mensajes de confirmación simultáneamente.	La pasarela de pago rechaza la transacción.	Este comportamiento proporciona protección frente a ataques de reenvío.
Realizar el proceso de compra en diferentes navegadores.	El sistema funciona correctamente en los navegadores que han sido comprobados: Google Chrome, Firefox e Internet Explorer.	Esta prueba muestra la operatividad del sistema en diferentes entornos.

Tabla 6.1: Pruebas realizadas al sistema

6.3. Conclusiones

Una vez realizada la prueba de conjunto se puede comprobar cómo el proceso de pago es realizado con éxito.

En los ficheros de depuración puede observarse cómo, dentro del módulo de autenticación mediante intercambio de mensajes SIP, se comprueba de manera satisfactoria la firma del mensaje y se verifica el origen de la misma al contrastar la información del comprador proporcionada por la pasarela de pago con la obtenida del mensaje de autorización del pago. Estas comprobaciones verifican la seguridad del sistema de verificación del pago a través IMS.

Por otro lado, las pruebas de seguridad realizadas en segunda instancia, prueban la robustez del sistema de pago frente a ataques de repetición, suplantación de identidad o interceptación. Además se ha comprobado la compatibilidad del sistema con varios navegadores, ya que la obtención de las cookies almacenadas en dichos navegadores constituye un punto clave en dicho sistema.

Capítulo 7

Historia del proyecto

El desarrollo del presente proyecto se ha realizado en el periodo de tiempo comprendido entre Octubre de 2009 a Junio de 2010.

Durante todos estos meses, se ha abordado el trabajo dividiéndolo en diferentes fases de desarrollo que se detallarán en el presente capítulo, identificando los objetivos establecidos para cada una de dichas fases, así como los problemas surgidos a lo largo de cada etapa y las estrategias tomadas para solucionarlos.

Las pruebas del sistema, descritas en el Capítulo 6, se fueron realizando de manera paralela al desarrollo del sistema, ya que para comprobar el correcto funcionamiento de la aplicación completa era necesario ir verificando el buen funcionamiento de cada uno de los módulos por separado.

7.1. Fases del proyecto

7.1.1. Fase 1: Familiarización con el entorno IMS

- **Descripción general:**

  Uno de los principales objetivos del presente proyecto era la provisión de servicio sobre la red IMS, estableciendo un caso de uso en el entrono de dicha red.

  Por ello, la familiarización con el entrono de red IMS, el manejo de sesiones, manipulación de aplicaciones cliente, registros en la red o la configuración de servidores de aplicaciones para su utilización en la red, debían ser manejadas y entendidas a la perfección antes de abordar el desarrollo del sistema propuesto.

  En esta fase se instalaron un núcleo de red IMS y un servidor de aplicaciones Sailfin. Se configuró la red para escuchar las peticiones provenientes y con destino a Sailfin para poder probar aplicaciones sencillas de comunicación entre terminales IMS y el AS.

- **Problemas:**

  Durante el período de realización de pruebas sobre el núcleo de red IMS, la mayor dificultad la encontré en la configuración de la red para poder acceder al AS desde ella, ya que para ello fue necesario crear un “trigger” y aplicar un filtro a las peticiones entrantes, además de realizar diferentes configuraciones en el DNS del ordenador.

- **Resultados:**

  Una vez realizadas las configuraciones comentadas anteriormente, se consiguió probar con éxito una aplicación de prueba que recibía peticiones provenientes de un terminal conectado a la red IMS y devolvía una respuesta al mismo.

7.1.2. Fase 2: Definición de requisitos

- **Descripción general:**

  Ante la necesidad de provisión de servicio a través de una aplicación convergente, se decidió la realización de una aplicación de comercio electrónico con pasarela de pago y soporte para autorización de dichos pagos mediante mensajería a través de IMS.

  La poca confianza existente en la actualidad por parte de los usuarios de Internet a proporcionar su información bancaria a través de la red, hizo que el desarrollo de un sistema de pago en el que la autenticación se realiza a través de dos canales de comunicación diferentes, además de la gran reputación social acerca de la seguridad en la telefonía móvil, resultara un tema interesante para tratar en el proyecto.

  Por otro lado, la necesidad de proveer un mecanismo de identificación centralizado y que evite al usuario la provisión de información no necesaria a los comercios se tomó como otro de los pilares básicos en torno a los que desarrollar el sistema.

- **Problemas:**

  La mayor preocupación en esta fase de desarrollo fue la búsqueda de estándares de identificación y autenticación interoperables entre sí y que se adaptasen correctamente a las necesidades del proyecto.

- **Resultados:**

  Se concretó el uso de OpenID como mecanismo de identificación y OpenSSO como marco para la provisión de autenticación debido a la sencillez para integrarlos. La selección de OpenID se fundamentó en el gran crecimiento experimentado en el uso y el desarrollo de dicho estándar en los últimos años, a pesar de que la integración en OpenSSO de otros estándares como por ejemplo SAML podría haberse realizado fácilmente también.

7.1.3. Fase 3: Implementación de un sistema de gestión de identidad mediante OpenID

- **Descripción general:**

  La provisión de identificación centralizada evitando al usuario la necesidad de crear diferentes perfiles de usuario para cada sitio Web que visite hizo que se soportase la identificación mediante OpenID en el sistema desarrollado.

  Para ello se desplegó un proveedor de identidad OpenID integrado en OpenSSO, que proporcionara los atributos requeridos por el comercio electrónico, mediante el uso de la librería OpenID4Java.

  En primer lugar se configuró el proveedor de identidad, estableciendo los atributos necesarios para el funcionamiento de la aplicación a desarrollar.

  Posteriormente se desarrolló el proveedor de servicio de identificación, realizando el descubrimiento de proveedor OpenID del identificador proporcionado por el usuario

para después redimirle a dicho IdP y recibir la respuesta del mismo una vez finalizado el intercambio de atributos.

- **Problemas:**

  El mayor problema encontrado fue la configuración del proveedor de identidad para establecer los atributos personalizados necesarios para la implementación del módulo de autenticación por teléfono móvil y la provisión de servicio de intercambio de atributos.

- **Resultados:**

  Finalmente se consiguió que el proveedor de servicios soportara la autenticación mediante OpenID 2.0 y el intercambio de atributos y se comunicara satisfactoriamente con el comercio electrónico.

7.1.4. Fase 4: Implementación de módulo de autenticación integrado en OpenSSO

- **Descripción general:**

  El desarrollo del módulo de autenticación integrado en OpenSSO se basó en la provisión de un mecanismo de verificación de transacciones económicas mediante el uso de un terminal IMS.

  Para ello, se estableció un funcionamiento de dicho módulo en diversos estados, en los que inicialmente el módulo envía la petición de confirmación al usuario para posteriormente quedar en estado de espera hasta que el usuario verifique el envío de dicho mensaje de confirmación del pago.

  Con el fin de verificar la firma digital de los mensajes provenientes de la aplicación cliente, se requirió el manejo de librerías para el uso de criptografía y firma digital en Java.

  Para la integración con IMS, el módulo debía retribuir las sesiones SIP existentes para poder poner comunicar el módulo con el terminal del usuario.

  Además se consideró necesario hacer uso de persistencia para controlar las peticiones realizadas por los usuarios con anterioridad.

- **Problemas:**

  La doble instanciación del siplet asociado al módulo de autenticación, como UA y como AS, hacían que la sesión SIP establecida al iniciarse la autenticación no pudiera obtenerse a partir de los parámetros SIPFactory y SIPSessionUtils.

- **Resultados:**

  Se solucionó el problema anteriormente descrito haciendo uso de un conjunto de clases de persistencia. Mediante el uso de dichas clases se estableció una base de

datos para el almacenamiento de las peticiones realizadas por los usuarios al comercio electrónico, permitiendo asociar las sesiones SIP establecidas a cada petición realizada y a las diferentes instancias del Siplet acceder a dicha base de datos.

7.1.5. Fase 5: Diseño de plugin para aplicación cliente

- **Descripción general:**

  En esta fase de desarrollo se implementó un plugin sobre la aplicación cliente "Monster the client", para la provisión de servicio de mensajería instantánea.

  Mediante el uso de dicho plugin, la aplicación ofrece al usuario la posibilidad de contestar automáticamente a las peticiones de confirmación de compra provenientes de la pasarela de pago mediante mensajes cifrados digitalmente.

  Para su implementación se hizo uso de las librerías proporcionadas por los desarrolladores de la aplicación.

- **Problemas:**

  El manejo de diferentes formatos de documentos, y la familiarización con las librerías para el uso de firma digital sobre documentos XML y de la aplicación cliente supusieron las mayores dificultades encontradas en esta fase.

- **Resultados:**

  Tras el análisis exhaustivo de las librerías de desarrollo se solventaron los problemas en la firma de los mensajes y el funcionamiento del plugin cumple los requisitos establecidos.

7.1.6. Fase 6: Integración del sistema

- **Descripción general:**

  Tras implementar los módulos de identificación y autenticación, además de la aplicación cliente, se integró el sistema con dos aplicaciones Web que gestionaran la interacción de dichos módulos.

  A pesar de que su desarrollo se realizó de forma paralela al de los módulos para ir comprobando su funcionamiento de forma progresiva, el diseño de las aplicaciones se ha refinado una vez finalizada la implementación de los mismos, dotando al comercio electrónico y a la pasarela de pago de un interfaz gráfico más agradable y fácil de usar.

- **Problemas:**

  La necesidad de intercambio de atributos entre las aplicaciones, tales como el importe o la descripción de la compra realizada, supuso un problema inicialmente, ya que resultaba difícil enviar atributos en redirecciones entre aplicaciones.

Posteriormente se solucionó la incidencia asociando los atributos a la sesión HTTP compartida por las aplicaciones.

- **Resultados:**

  La ejecución de la aplicación completa fue posible una vez finalizada esta fase, identificando al usuario una vez realizada la compra, redirigiéndole más tarde a su pasarela de pago y verificando la transacción mediante su terminal IMS.

7.1.7. Fase 7: Documentación y pruebas

- **Descripción general:**

  Esta fase consiste en la documentación del trabajo realizado y la realización de esta memoria.

7.2. Opinión personal

La posibilidad de realizar un estudio acerca de un tema de tal importancia como la verificación de las transacciones en los comercios electrónicos me ha resultado muy interesante y me ha creado un gran interés acerca del estudio de la gestión de identidad y los mecanismos de seguridad en la red.

Durante la realización del proyecto, la mayor dificultad la encontré al manejar una gran variedad de conceptos tan dispares como comercio electrónico, gestión de identidad o manejo de redes de próxima generación, para ensamblarlos en un mismo sistema.

Por otro lado, el manejo de Servlets SIP y la recuperación de las sesiones SIP establecidas, así como del contexto SIP de la aplicación, resultaron otros de los puntos de mayor dificultad durante el desarrollo.

En cuanto a experiencia personal, el presente proyecto me ha aportado gran cantidad de conocimientos y experiencia acerca de los medios de pago electrónicos, la gestión de identidad, y la programación Web, a la cual no estaba muy acostumbrado.

Capítulo 8

Conclusiones

8.1. Conclusiones

El sistema desarrollado en el presente proyecto se ha orientado a desarrollar un mecanismo de pago consistente en una pasarela de pago que permita la realización de transacciones económicas a través de una red IMS haciendo uso de dos canales de comunicación diferenciados: Internet y la red de telefonía móvil.

El uso de Internet es cada vez más frecuente en la realización de transacciones económicas y por ello la provisión de un servicio eficaz, robusto, fiable y adaptado a las nuevas tecnologías resulta fundamental.

Sin embargo, el rechazo existente entre los usuarios de Internet a proporcionar su información bancaria a través de la red hacen que se haya planteado el desarrollo de un método de pago que proporcione una mayor seguridad a los mismos, a la par que se adapte al desarrollo experimentado por las redes de comunicaciones.

Por ello, se ha realizado el desarrollo de un mecanismo de autenticación de usuarios asociado a la confirmación de transacciones mediante el uso de dos canales de comunicación. La utilización de ambos canales favorece la implementación de un servicio fiable y robusto ante ataques del exterior, ya que resulta muy complicado comprometer ambas redes de manera simultánea.

El módulo de verificación de pagos desplegado provee al sistema de una gran robustez frente a fraudes electrónicos. Mediante la pasarela de pago se protege la información referida a la transacción económica entre el comercio electrónico y al usuario al intercambiar dicha información cifrada y a través de un canal alternativo como es IMS.

La gestión de seguridad en el intercambio de los documentos de autorización del pago se ha realizado mediante cifrado y firma digital. Además, el sistema desarrollado verifica la identidad del origen del mensaje de confirmación de compra, lo que dota al sistema de protección frente a la suplantación de identidad del cliente, así como los ataques de intercepción o reenvío.

El desarrollo del mecanismo de pago seguro desarrollado estará orientado a motivar tanto a usuarios como a comerciantes a potenciar el uso del comercio electrónico y a aumentar su grado de confianza en la aplicación de las nuevas tecnologías a las transacciones económicas a través de la red.

Adicionalmente al servicio de pago seguro, se ha diseñado una aplicación cliente que facilita la verificación de transacciones a los usuarios, lo que favorece a la disminución del tiempo empleado en la realización de compras electrónicas, así como proporciona una mayor comodidad al usuario para verificar dichas compras.

Por otro lado, se ha dotado al sistema de un sistema de gestión de identidad, por que se evita al usuario la obligación de registrase en las diferentes páginas Web a las que acceda, lo que anteriormente suponía una gran pérdida de tiempo y la vulneración de la privacidad de dichos usuarios.

Dicho módulo de gestión de identidad proporciona al sistema de la capacidad para proteger la identidad digital asociada a los usuarios, permitiendo que ellos controlen cuándo proporcionan información, a quién y de qué tipo. Una vez implementado el sistema de gestión de identidad, el usuario podrá tener acceso a diferentes proveedores de servicio haciendo uso de las mismas credenciales.

Aun así, el servicio de mayor interés proporcionado por el presente proyecto es la integración de los diferentes módulos que lo componen. Es decir, la integración de la gestión de identidad del usuario con la provisión de autenticación a través de una aplicación convergente, protegiendo la identidad digital de dicho usuario a la par que proporcionándole un mecanismo para el pago seguro en sus transacciones electrónicas.

La adaptación de los actuales mecanismos de pago, como las pasarelas de pago, a las redes de próxima generación, tales como IMS, ofrecen una visión más realista de las oportunidades de desarrollo proporcionadas por dichas redes. Además, la implementación de ejemplos de provisión de servicio sobre las mismas hace que la migración hacia el concepto de red convergida sea aceptada de manera más sencilla y gradual.

8.2. Líneas futuras de investigación

En este apartado se presentan las diferentes líneas de trabajo que se podrían seguir sobre el tema tratado en el proyecto. A continuación detallamos alguna de ellas:

- Integración de varias tecnologías de gestión de identidad:

  Podría añadirse al comercio electrónico la posibilidad de identificar al usuario mediante otra especificación para la gestión de identidad, como por ejemplo SAML, que al igual que OpenID es integrable en OpenSSO y proporcionaría una mayor versatilidad a la aplicación al permitir al sistema gestionar la identidad del usuario mediante diferentes alternativas.

- Extensión de la funcionalidad del comercio electrónico:

  El desarrollo de la aplicación de comercio electrónico ha sido bastante básico y orientado a la demostración del funcionamiento de un proveedor de servicio sobre la red IMS. Por ello, se podría ampliar la capacidad de dicha aplicación, por ejemplo, dotándola de una base de datos que gestionara los pedidos de los usuarios para así implementar un carrito de la compra.

- Extensión de la funcionalidad del módulo de autenticación:

  El módulo de autenticación, tal y como se encuentra diseñado, puede almacenar las veces que un usuario determinado ha intentado realizar una compra sin éxito, pero en

la versión actual no reacciona frente a este hecho. Por tanto, podrían gestionarse la cancelación de la compra al alcanzarse cierto número de intentos.

A la hora de realizar el diseño del módulo de autenticación se establecieron dos estados adicionales a los descritos en este texto que no han sido finalmente implementados, relacionados con la gestión de terminales SIP y usuarios registrados en el módulo. La implementación de dichos estados y la lógica asociada a los mismos dotaría al sistema de una mayor robustez y fortaleza frente a ataques de suplantación de identidad.

- Extensión de la funcionalidad de la pasarela de pago:

  Podría mejorarse el comportamiento de la pasarela de pago ante los hechos detallados en 6.2, reaccionando ante posibles fallos del usuario, por ejemplo, al pulsar el botón “Done” antes de haber enviado el mensaje de confirmación del pago.

Apéndice A

Presupuesto

En este apartado se detallará el presupuesto asociado a la realización del presente proyecto, detallando los costes de personal que ha desarrollado el mismo, así como del material utilizado para dicho fin. La suma de estos diversos costes constituirá el presupuesto final de dicho proyecto.

A.1. Costes de personal

El coste de personal estará principalmente asociado al sueldo de un Ingeniero de Telecomunicación encargado del proyecto en cuestión. Según los datos obtenidos del COIT[46], el salario de un Ingeniero de Telecomunicación se situaba en torno a los 72 euros por hora en 2008. Actualizando dicho salario según el crecimiento del IPC en los años 2009 y 2010 (1.15 según las predicciones marcadas en [47]), alcanza el valor de 72.82 euros por hora.

El desarrollo del proyecto, realizando una media sobre los días empleados en la realización del mismo, ha supuesto el empleo de aproximadamente 4 horas al día durante el período comprendido entre Septiembre de 2009 y Junio de 2010, un total de 10 meses.

Considerando que el número medio de días laborables en un mes es de 20 días, el total de horas empleadas en el desarrollo mencionado asciende a 800 horas, siendo el coste total del personal 58.256 euros. La Tabla A.1 recoge este resultado.

Concepto	Horas	Honorarios	Importe
Ingeniero de Telecomunicación	800 horas	72 €/hora	58.256 €

Tabla A.1: Costes de personal

A.2. Costes de material

A continuación se detalla el material utilizado en el desarrollo del proyecto:

- Ordenador portátil con sistema operativo Linux y conexión inalámbrica, valorado en 700 euros.
- Software de libre distribución.
- Conexión a Internet para obtener la información necesaria para la implementación de los diferentes módulos del proyecto, realizar conexión remota a las máquinas de la Universidad y desplegar el sistema desarrollado en el servidor de aplicaciones. Supone un gasto de unos 40 euros al mes, originando un total de 400 euros.

La siguiente tabla muestra el resumen de los costes de material del proyecto.

Concepto	Unidades	Precio/unidad	Importe
Ordenador portátil	1	700 €	700 €
Conexión a Internet	1	400 €	400 €
Total			1.100 €

Tabla A.2: Costes de material

A.3. Presupuesto total

Como se indicaba anteriormente, el presupuesto total del proyecto estará formado por la suma de los costes de personal y los costes de material detallados anteriormente. La Tabla A.3 muestra los resultados obtenidos.

Concepto	Importe
Costes de personal	58.256 €
Costes de material	1.100 €
Total	59.356 €

Tabla A.3: Presupuesto total del proyecto

Apéndice B

Manual de instalación

B.1. Configuración del módulo de gestión de identidad mediante OpenID en OpenSSO

A continuación se detallan las configuraciones que se deben realizar para poner en funcionamiento la implementación del módulo de gestión de identidad según la especificación OpenID, integrada en OpenSSO.

- Configuración del módulo OpenID:
  - Cargar el archivo `openid.war` obtenido de OpenSSO en el AS.
  - Actualizar los ficheros `AMConfig.properties`, `Provider.properties` y `ldap.properties` con los valores tomados de la implementación OpenSSO.

    El detallado del contenido de estos archivos de configuración se escapa de los objetivos de este texto, por lo que simplemente se destacará la necesidad de introducir los valores de atributos que se desea que contenga el perfil OpenSSO/OpenID del usuario en los mismos.
  - Añadir los ficheros de propiedades al directorio de clases y reiniciar el dominio.
- Configuración de OpenSSO:
  - Enlazar el módulo OpenID a la lista de dominios de confianza.
  - Añadir un atributo OpenID al esquema de usuario OpenSSO.
  - Añadir dicho atributo al directorio LDAP embebido en OpenSSO.
  - Habilitar la posibilidad de actualización del atributo OpenID desde el directorio LDAP.
  - Añadir los atributos del perfil de usuario, configurando LDAP para que sean accesibles desde OpenID.
  - Reiniciar el AS.

B.2. Configuración del módulo de autenticación

Para configurar el módulo de autenticación, debemos poseer varios ficheros:

- `AMAuthIMS.jar`: Fichero que contiene las clases que implementan el módulo de autenticación propiamente dicho.

- `AMAuthIMS.xml`: Fichero de configuración del módulo de autenticación que define los estados que presentará el módulo de autenticación y los requerimientos de cada uno de ellos.

En primer lugar, deberemos situarnos en el directorio donde se encuentre instalado OpenSSO. Una vez allí, debemos copiar el fichero `AMAuthIMS.jar` a `WEB-INF/lib.`

Posteriormente, accedemos al directorio `/config/auth/default` y copiamos el fichero de configuración `AMAuthIMS.xml`:

A continuación, accedemos a la página de configuración de módulos de autenticación, y nos logueamos como administrador.

Entramos en Configuración -> Autenticación -> Principal y añadimos una nueva clase de módulo de autenticación conectable de nombre: `es.uc3m.it.pervasive.authModule.AMAuthIMS.` Pulsamos el botón de "Guardar", para salvar la configuración establecida.

Para finalizar el proceso reiniciamos OpenSSO y el módulo ya estará disponible.

B.3. Instalación del plugin de mensajería en la aplicación cliente

A continuación se detalla el proceso de instalación del módulo de mensajería instantánea en la aplicación cliente "Monster the client".

En primer lugar, debemos tener instalado en nuestro ordenador la aplicación "Monster the client". Para ello accedemos a [45] y pulsamos en la pestaña "Downloads". Descargamos el fichero que encontramos allí, lo descomprimimos.

Para ejecutar el programa abrimos un terminal, nos situamos en la carpeta que hemos descomprimido anteriormente y ejecutamos el siguiente comando:

```
sh monster
```

El primer paso a realizar es la configuración del perfil de usuario para la conexión a una red IMS. Para ello, en la pantalla de inicio de la aplicación seleccionamos la opción "Add Profile".

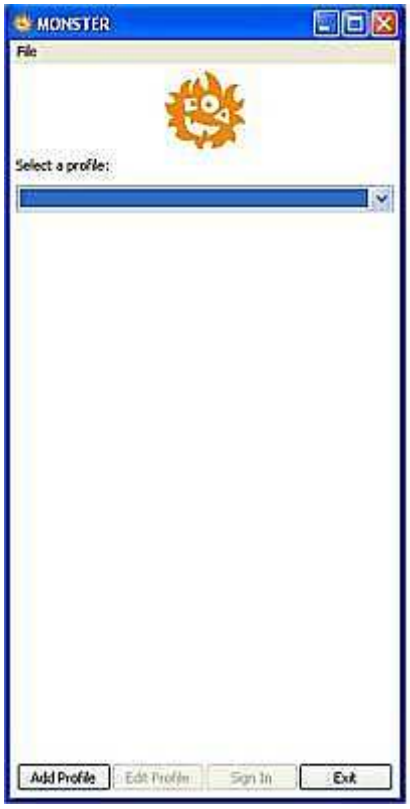

Figura B.1: Añadir perfil de usuario

Posteriormente deberán introducirse los datos que mostramos en la Figura B.2 en la pantalla de configuración de perfil.

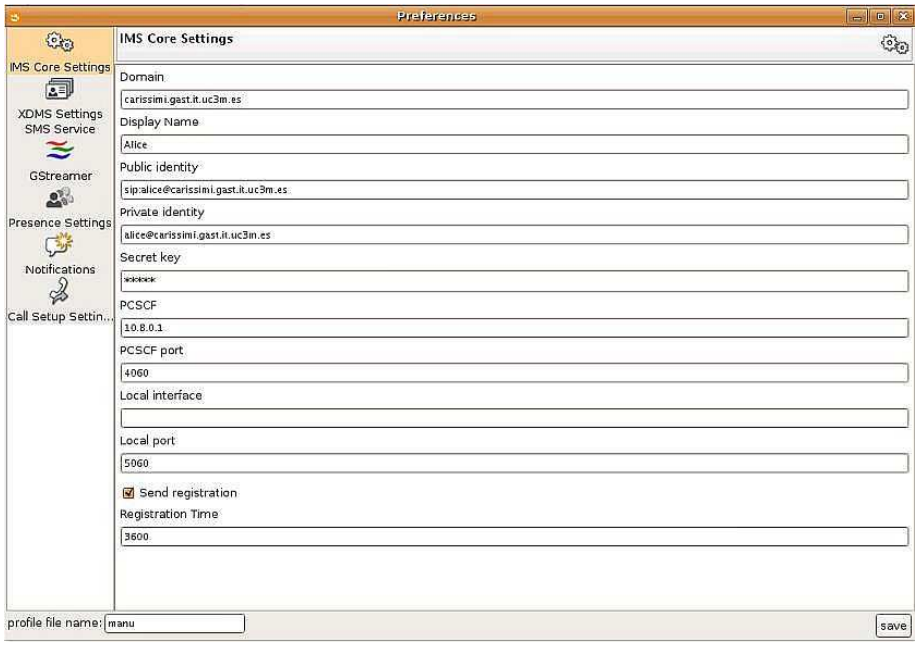

Figura B.2: Configuración de perfil de usuario

Una vez creado el perfil de usuario, cerramos la aplicación y procedemos a la carga del plugin diseñado.

Para ello, copiamos el fichero `addon-SMS.jar`, que contiene las clases que implementan al módulo, a la carpeta donde se encuentra la aplicación “Monster the client”, es decir, la carpeta que descomprimimos al inicio de esta sección.

Una vez copiado dicho archivo al directorio en cuestión, al volver a ejecutar la aplicación, el módulo de mensajería automática ya estará disponible.

Apéndice C

Glosario de términos

Atributo: Es la información asociada a una entidad y que provee información acerca de una característica de la misma. En el marco de la gestión de identidad y el de la autenticación el perfil de usuario contendrá dichos atributos.

Autenticación: Es el acto de confirmación de la autenticidad de una entidad y, en el caso concreto de las personas, suele estar relacionado con la confirmación de la identidad de las mismas.

Autorización: Es el acto de proteger los recursos de un determinado sistema, para que su uso se encuentre limitado a los consumidores autorizados para tal fin.

Certificado: Conjunto de datos de seguridad relevantes emitidos por una autoridad de seguridad o un tercero de confianza. Son emitidos para proporcionar servicios de integridad y autenticación.

Círculo de confianza: Es un conjunto de criterios dentro de una federación para permitir el acceso fiable a los recursos de las diferentes organizaciones asociadas a ella.

COIT: Colegio Oficial de Ingenieros de Telecomunicación.

Control de acceso: Es la técnica orientada a evitar el uso de un recurso de manera no autorizada.

Contexto: Son las circunstancias bajo las que un dispositivo o entidad está siendo utilizados, y la información asociada a dichas circunstancias.

Criptografía: Es la ciencia de cifrar y descifrar información mediante técnicas especiales normalmente orientada al intercambio de mensajes que sólo puedan ser leídos por personas a las que vayan dirigidos y que posean los medios para descifrarlos.

Descubrimiento: Es el proceso mediante el cual los recursos de un sistema de gestión de identidad son localizados.

EMV: Es un estándar de interoperabilidad de tarjetas IC ("Tarjetas con chip") y TPV con soporte IC, para la autentificación de pagos mediante tarjetas de crédito y débito.

Entidad: Es una persona o dispositivo que se caracteriza por sus atributos.

Federación: Es la asociación establecida entre varios proveedores de servicio y proveedores de identidad.

Firma digital: Es un método criptográfico que asocia la identidad de una persona al mensaje o documento que ésta envía, pudiendo asegurar la integridad del mismo.

GlassFish: Es un servidor de aplicaciones de código libre y gratuito desarrollado por Sun Microsystems que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación.

HTTP: HyperText Transfer Protocol. Es un protocolo de transferencia de hipertexto pata transacción de mensajes según esquema de petición-respuesta entre cliente y servidor.

Identidad federada: Es la identidad local de un usuario que permite acceder a los servicios o aplicaciones asociadas a una federación determinada.

IdP: Proveedor de identidad (Identity Provider). Es una entidad que emite, gestiona y mantiene entidades digitales fiables para otras entidades.

Java EE: Java Enterprise Edition es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje Java con arquitectura de N niveles distribuidos, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

JSF: Java Server Faces [48] es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

JSP: Java Server Pages es una tecnología y marco para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JSP como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL.

IMS: El Subsistema Multimedia IP (o IP Multimedia Subsystem) forma parte del núcleo de la arquitectura de las redes de siguiente generación. Estas redes son capaces de proporcionar servicios multimedia fijos y móviles.

NGN: Red de Siguiente Generación o Red Próxima Generación (Next Generation Networking) es un amplio término que se refiere a la evolución de la actual infraestructura de redes de telecomunicación y acceso telefónico con el objetivo de lograr la congruencia de los nuevos servicios multimedia (voz, datos, video...) en los próximos 5-10 años.

OpenID: Es un estándar de identificación digital descentralizado, con el que un usuario puede identificarse en una página web a través de una URL (o un XRI en la versión actual) y puede ser verificado por cualquier servidor que soporte el protocolo.

OpenSSO: Es una plataforma de código abierto para la federación de servicio y el control de acceso, permitiendo a las aplicaciones que hagan uso de él permitirán a sus usuarios proporcionar sus datos de identificación una sola vez y quedar identificado en el resto de aplicaciones integradas en el sistema

Perfil de usuario: Es el conjunto de atributos y variables asociadas a un determinado usuario en un marco determinado.

Phising: Es un término informático que denomina un tipo de delito encuadrado dentro del ámbito de las estafas cibernéticas, y que se comete mediante el uso de un tipo de ingeniería social caracterizado por intentar adquirir información confidencial de forma fraudulenta [49](como puede ser una contraseña o información detallada sobre tarjetas de crédito u otra información bancaria)

Pharming: Es la explotación de una vulnerabilidad en el software de los servidores DNS (*Domain Name System*) o en el de los equipos de los propios usuarios, que permite a un atacante redirigir un nombre de dominio a otra máquina distinta.

Principal: Entidad susceptible de ser autenticada.

RP: Relying Party es la entidad que actúa de intermediaria entre el proveedor de servicio y el proveedor de identidad en un sistema de gestión de identidad, procesando las peticiones y las respuestas intercambiadas entre ambos.

RSA: (Rivest, Shamir y Adleman) es un sistema criptográfico de clave pública desarrollado en 1977. En la actualidad, RSA es el primer y más utilizado algoritmo de este tipo y es válido tanto para cifrar como para firmar digitalmente.

Sailfin: Es un servidor de aplicaciones de código libre y gratuito desarrollado por Sun Microsystems que implementa la especificación JCP SIP Servlet 1.1 (JSR 289) integrándola en GlassFish.

SET: *Secure Electronic Transaction*. Es un protocolo estándar para proporcionar seguridad a una transacción con tarjeta de crédito en redes de ordenadores inseguras, en especial Internet.

SIP: Session Initiation Protocol. Protocolo orientado a la a iniciación, modificación y finalización de sesiones interactivas de usuario donde intervienen elementos multimedia como vídeo, mensajería instantánea, voz o realidad virtual.

SSO: Single Sign-On. Es un procedimiento de autenticación que habilita al usuario para acceder a varios sistemas con una sola instancia de identificación.

TPV: Es el acrónimo de Terminal de Punto de Venta. Hace referencia al dispositivo y tecnologías que ayudan en las tareas de gestión de un establecimiento comercial de venta al público.

X.509: Es un estándar UIT-T para infraestructuras de clave pública (*Public Key Infrastructure* o PKI). X.509 especifica, entre otras cosas, formatos estándar para certificados de clave pública y un algoritmo de validación de la ruta de certificación.

XML: Extensible Markup Language. Es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C) que proporciona una manera de definir lenguajes para diferentes necesidades.

XRI: (Extensible Resource Identifier) es un nuevo sistema de identificación en Internet, diseñado específicamente para identidades digitales de dominio cruzado.

Bibliografía

[1] Zhang, Guo-Qing. “Evolution of the Internet and its Cores”. Diciembre 2008.

[2] Staunton, Clare. “E-commerce across Europe. Progress and prospects”. Octubre 2008.

[3] EMV. EMV 2000 Integrated Circuit Card Specification for Payment Systems Version 4.0 - Book 1: Application Independent IC Card to Terminal Interface Requirements. EMVCo, 2000.

[4] Página Web oficial de 4B. Disponible en www.4b.es. Junio 2010.

[5] Cyveillance. “The Cost of Phishing: Understanding the True Cost Dynamics Behind Phishing Attacks.” Disponible en http://www.cyveillance.com/. Octubre de 2009.

[6] S. Subenthiran, Dr.K.Sandrasegaran, R.Shalak. “Requirements for Identity Management in Next Generation Networks”. Julio 2009.

[7] Gonzalo Camarillo, Miguel-Angel García-Martín. "The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds" (John Wiley & Sons). 2006.

[8] Luis Martínez López, Francisco Mata Mata, Enrique Bernal Jurado. “Medios de pago electrónico. Piedra angular en el desarrollo del comercio electrónico.” Disponible en http://sinbad2.ujaen.es/sinbad2/files/publicaciones/77.pdf. Junio 2010.

[9] Javier Santomá Juncadella. ”Nuevos medios de pago electrónico: Hacia la desintermediación bancaria. Febrero 2004.

[10] X.509 Public Key Infrastructure Specification. RFC 2459. Disponible en http://www.ietf.org/rfc/rfc2459.txt.

[11] SETCo. Secure Electronic Transaction Specification – Bookks 1-4. SETCo, 1997.

[12] Vorapranee Khu-Smith, Chris J. Mitchell. ”Using EMV cards to protect e-commerce transactions”. Disponible en http://www.isg.rhul.ac.uk/cjm/uectpe.pdf. 2008.

[13] Sociedad de la información. “Pasarelas y medios de pago electrónico”.Disponible en http://sinbad2.ujaen.es/sinbad2/files/publicaciones/77.pdf. Enero 2006.

[14] Página Web oficial de Euro 6000. Disponible en http://www.euro6000.es. Junio 2010.

[15] Common Electronic Purse Specifications (CEPSCO). Functional Requirements, Version 6.3. Disponible en http://www.irisa.fr/vertecs/Equipe/Rusu/FME02/functionalrequirements6-3.pdf. Septiembre 1999.

[16] Página Web de Valimo. Disponible en http://www.valimo.com/. Junio 2010.

[17] Página Web de Trivnet. Disponible en http://www.trivnet.com/. Junio 2010.

[18] Identity and Access control. Apuntes de Computación Ubicua. Master Interuniversitario en Ingeniería Telemática de la Universidad Carlos III de Madrid. Abril 2010.

[19] OSCON 2005 Keynote. Identity 2.0. Disponible en http://identity20.com/media/OSCON2005/. Diciembre 2005.

[20] Lucent Technologies & Sun Microsystems. "Identity Management for Converged Networks". February 2006.

[21] S.Gálvez Rojas, J. Lago Cabrera, J.L. de la Rosa Triviño, C. González Florido, D. Palacios Jiménez, E. Gutiérrez Marín, J.A. Ruiz Moreno, R. García Hermoso. "OpenSSO y OpenID. Comparativa y capacidad de integración". Disponible en http://www.mundointernet.es/IMG/pdf/ponencia151_2.pdf. Junio 2010.

[22] Página oficial de OpenSSO. Disponible en https://opensso.dev.java.net/. Junio 2010.

[23] OpenID Provider Authentication Policy Extension 1.0 Specification. Disponible en http://openid.net/specs/openid-provider-authentication-policy-extension-1_0.html. Diciembre 2008.

[24] OpenID Authentication 2.0 Specification. Disponible en http://openid.net/specs/openid-authentication-2_0.html. Diciembre 2007.

[25] OpenID Attribute Exchange 1.0 Specification. Disponible en http://openid.net/specs/openid-attribute-exchange-1_0.html. Diciembre 2007.

[26] Miller, J., "Yadis Specification 1.0". 2005.

[27] OpenID.net "Start Using your OpenID". Disponible en http://openid.net/get-an-openid/start-using-your-openid/. Junio 2010.

[28] XML Signature Syntax and Processing (Second Edition), W3C Recommendation. Disponible en http://www.w3.org/TR/xmldsig-core/. Junio 2008.

[29] Página oficial de RSA Laboratories. http://www.rsa.com/rsalabs/node.asp?id=2146. Junio 2010.

[30] Goodies4you.com "RSA encryption explained simply". Disponible en http://www.goodies4uall.com/rsa-explained-simply/cprogramming/. Junio 2010.

[31] 3GPP Specifications. Disponible en http://www.3gpp.org/specifications. Junio 2010.

[32] Session Initiation Protocol (RFC 3261). Disponible en http://www.ietf.org/rfc/rfc3261.txt. Junio 2002.

[33] Daniel Diaz Sanchez, Andrés Marín López, Patricia Arias Cabarcos "Arquitectura de redes". Junio 2010.

[34] Página oficial de GlassFish. Disponible en https://glassfish.dev.java.net/. Junio 2010.

[35] Página oficial del proyecto Sailfin. Disponible en https://sailfin.dev.java.net/. Junio 2010.

[36] Librerías OpenID. Disponible en http://wiki.openid.net/Libraries. Septiembre 2009.

[37] S. Hamid y J. Stepka. ”Using OpenID”. Disponible en http://www.theserverside.com/tt/articles/article.tss?l=OpenID. 2007.

[38] Matamoros Casas, Teresa “PAPOID PAPI OpenID Server”. Disponible en http://www.rediris.es/ptyoc/res/store/dl20/PAPOID-PFC.pdf. Junio 2008.

[39] Implementación OpenID4Java. Disponible en http://code.google.com/p/openid4java/. Junio 2010.

[40] Tutorial Java para generación y verificación de firmas digitales. Disponible en http://java.sun.com/docs/books/tutorial/security/apisign/gensig.html. Junio 2010.

[41] Custom OpenSSO AUthentication Modules. Disponible en http://blogs.sun.com/docteger/entry/custom_auth_module_opensso. Junio 2010.

[42] Oracle. Integrating applications with OpenSSO. Disponible en http://developers.sun.com/identity/reference/techart/app-integration.html. Junio 2010.

[43] Universidad de Sevilla, Integración de aplicaciones en OpenSSO. Disponible en https://opensso.us.es/integracion/. 2009/10.

[44] Página Web oficial de Java Server Pages . http://java.sun.com/products/jsp/. Junio 2010.

[45] Página oficial de Monster The Client. Disponible en http://www.monster-the-client.org/. Junio 2010.

[46]Página web del COIT. Disponible en http://www.coit.es. Junio 2010.

[47] Blog Financiero “De finanzas". Disponible en http://definanzas.com/2009/01/23/ipc-2009/. Junio 2010.

[48] Página Web oficial de Java Server Faces. http://java.sun.com/javaee/javaserverfaces/. Junio 2010.

[49] Cyveillance. The Cost of Phishing: Understanding the True Cost Dynamics Behind Phishing Attacks. Editado por http://www.cyveillance.com/.Octubre 2008.

[50] Antonio Valverde García. “Seguridad en los nuevos medios de pago”. Disponible en http://www.iec.csic.es/CRIPTONOMICON/articulos/expertos30.html. 2000.

[51] E. Bond. “Securing the Blogosphere Through OpenID: An Introduction”. Disponible en http://dev.aol.com/article/2007/05/openid_blog. Mayo 2007.

[52] SIP programming for the Java developer. Disponible en http://www.javaworld.com/javaworld/jw-06-2006/jw-0619-sip.html?page=4. Junio 2010.

[53] Wikipedia. Disponible en http://es.wikipedia.org/. Junio 2010.