Tecnología Blockchain Aplicada a la Securización e Interoperabilidad de eSIMs en Dispositivos IoT

Blockchain Technology Applied to the Security and Interoperability of eSIMs in IoT Devices

Trabajo de Fin de Máster
Curso 2024 - 25

Autor
Johan Molina Medina

Director
Samer Hassan Collado

Master en Internet de la Cosas
Facultad de Informática
Universidad Complutense de Madrid

Tecnología Blockchain Aplicada a la Securización e Interoperabilidad de eSIMs en Dispositivos IoT

Blockchain Technology Applied to the Security and Interoperability of eSIMs in IoT Devices

Trabajo de Fin de Master en Internet de las Cosas
Departamento de Ingeniería de Software e Inteligencia Artificial

Autor
Johan Molina Medina

Director
Samer Hassan Collado

Convocatoria: 04/07/2025
Calificación: 9

Master en Internet de la Cosas
Facultad de Informática
Universidad Complutense de Madrid

04 de Julio de 2025

Dedicatoria

A mi hijo Matías, inspiración constante y motor de cada uno de mis esfuerzos.
A mi esposa Ingrid, por su paciencia y apoyo incondicional.
A toda mi familia, que, a pesar de la distancia, me ha acompañado con su aliento y confianza en cada paso de este camino.
A mis amigos, por estar siempre presentes con una palabra de ánimo y por recordarme que nunca estoy solo.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a mis compañeros del máster, con quienes compartí experiencias, desafíos y aprendizajes que enriquecieron profundamente este proceso.

A los profesores del programa, por su dedicación, compromiso y generosidad al compartir su conocimiento. Sin ustedes, este trabajo no habría sido posible.

De manera muy especial, mi agradecimiento al profesor Samer Hassan, mi tutor, por su orientación, paciencia y constante apoyo, especialmente en los momentos más difíciles. Su guía fue clave para la culminación de este proyecto.

A todos los que, de una u otra forma, han contribuido a este logro, les extiendo mi más profundo agradecimiento.

Resumen

Tecnología Blockchain Aplicada a la Securización e Interoperabilidad de eSIMs en Dispositivos IoT

El crecimiento exponencial del Internet de las Cosas (IoT) y la necesidad de una conectividad flexible han posicionado a la tecnología eSIM (Embedded Subscriber Identity Module) como un pilar fundamental. Sin embargo, su gestión sigue anclada en modelos centralizados controlados por un único operador o proveedor, lo que genera una fricción operativa significativa, limita la verdadera interoperabilidad entre actores y carece de un registro unificado y auditable del ciclo de vida de los dispositivos. Esta falta de transparencia y la dependencia de intermediarios representan una barrera para la escalabilidad y la confianza en un ecosistema que demanda agilidad y seguridad.

Este trabajo aborda directamente estos desafíos proponiendo un modelo innovador que utiliza la tecnología blockchain para crear un marco de gestión descentralizado. Se diseñan dos arquitecturas técnicas donde un componente clave, el IoT Profile Assistant (IPA), gestiona los perfiles de conectividad del dispositivo. En la primera arquitectura, el IPA se aloja en el dispositivo IoT, y en la segunda, se integra en la eUICC (Embedded Universal Integrated Circuit Card), reforzando aún más la seguridad. Ambas arquitecturas se orquestan mediante un backend y utilizan contratos inteligentes (smart contracts) para registrar de forma inmutable y transparente eventos críticos como la activación de perfiles, los cambios de operador y las operaciones de autenticación.

La viabilidad de la solución fue demostrada mediante una prueba de concepto técnica en un entorno controlado y validada cualitativamente a través de una encuesta a 57 profesionales del sector. Los resultados concluyen que la integración de blockchain mejora significativamente la securización, trazabilidad y auditabilidad, ofreciendo un marco de gestión de eSIMs más resiliente y automatizable. De este modo, el trabajo no solo presenta una solución técnica, sino que sienta las bases para un ecosistema IoT más abierto, interoperable y confiable, alineado con las futuras normativas regulatorias y modelos de gobernanza distribuida.

Palabras claves

API, Backend, Blockchain, eUICC, IoT, GSMA,

Abstract

Blockchain Technology Applied to the Security and Interoperability of eSIMs in IoT Devices

The exponential growth of the Internet of Things (IoT) and the need for flexible connectivity have positioned eSIM (Embedded Subscriber Identity Module) technology as a fundamental pillar. However, its management remains anchored in centralized models controlled by a single operator or provider, which generates significant operational friction, limits true interoperability between actors, and lacks a unified and auditable record of the device lifecycle. This lack of transparency and reliance on intermediaries represents a barrier to scalability and trust in an ecosystem that demands agility and security.

This work directly addresses these challenges by proposing an innovative model that leverages blockchain technology to create a decentralized management framework. Two technical architectures are designed, where a key component, the IoT Profile Assistant (IPA), manages the device's connectivity profiles. In the first architecture, the IPA is hosted on the IoT device, and in the second, it is integrated within the eUICC (Embedded Universal Integrated Circuit Card), further enhancing security. Both architectures are orchestrated by a backend and use smart contracts to immutably and transparently record critical events such as profile activations, operator changes, and authentication operations.

The feasibility of the solution was demonstrated through a technical proof of concept in a controlled environment and qualitatively validated through a survey of 57 professionals in the sector. The results conclude that the integration of blockchain significantly improves security, traceability, and auditability, offering a more resilient and automatable eSIM management framework. In this way, the work not only presents a technical solution but also lays the foundation for a more open, interoperable, and trustworthy IoT ecosystem, aligned with future regulatory frameworks and distributed governance models.

Keywords

API, Backend, Blockchain, eUICC, IoT, GSMA, Smart contracts.

Índice de contenidos

Índice de figuras

Glosario

- **APDU (Application Protocol Data Unit):** Protocolo de comunicación de bajo nivel utilizado para el intercambio de comandos con tarjetas inteligentes, como la eUICC.
- **API (Application Programming Interface):** Interfaz de Programación de Aplicaciones; conjunto de reglas y herramientas que permiten que diferentes aplicaciones de software se comuniquen entre sí.
- **ASN.1 (Abstract Syntax Notation One):** Estándar para describir y codificar datos utilizado en telecomunicaciones y criptografía para garantizar la interoperabilidad.
- **BSS (Business Support System):** Sistema de Soporte al Negocio; componentes utilizados por los operadores para gestionar procesos orientados al cliente, como la facturación y la gestión de pedidos.
- **CoAP (Constrained Application Protocol):** Protocolo RESTful diseñado por la IETF para dispositivos y redes con recursos restringidos, como las redes IoT.
- **CRL (Certificate Revocation List):** Lista de certificados digitales que han sido revocados por la Autoridad de Certificación (CA) antes de su fecha de expiración.
- **DTLS (Datagram Transport Layer Security):** Protocolo que proporciona seguridad a las comunicaciones basadas en datagramas (como UDP), garantizando confidencialidad e integridad.
- **ECC (Elliptic Curve Cryptography):** Criptografía de Curva Elíptica; un enfoque de criptografía de clave pública que ofrece un nivel de seguridad equivalente a RSA con claves más pequeñas y mayor eficiencia.
- **ECDSA (Elliptic Curve Digital Signature Algorithm):** Algoritmo de Firma Digital de Curva Elíptica, ampliamente utilizado en blockchains como Ethereum para firmar transacciones.
- **EID (eUICC Identifier):** Identificador único de la eUICC; número de serie que identifica de forma exclusiva a una tarjeta eUICC.
- **eIM (eSIM IoT Remote Manager):** Gestor Remoto de eSIM para IoT; entidad lógica que actúa como intermediario para facilitar la gestión remota de perfiles eSIM en dispositivos IoT.
- **eSIM (Embedded Subscriber Identity Module):** Módulo de Identidad de Abonado Integrado; una versión digital y programable de la tarjeta SIM tradicional, integrada directamente en el hardware del dispositivo.

- **eUICC (Embedded Universal Integrated Circuit Card):** Tarjeta de Circuito Integrado Universal Embebida; el componente de hardware seguro (chip) donde se almacenan y gestionan los perfiles de eSIM.
- **EVM (Ethereum Virtual Machine):** Máquina Virtual de Ethereum; el entorno de ejecución en el que se ejecutan los contratos inteligentes en la red Ethereum y otras compatibles.
- **GDPR (General Data Protection Regulation):** Reglamento General de Protección de Datos de la Unión Europea.
- **GSMA (Global System for Mobile Communications Association):** Asociación global de operadores móviles que define los estándares para la tecnología móvil, incluyendo la eSIM.
- **HMAC (Hash-based Message Authentication Code):** Código de autenticación de mensajes basado en hash; mecanismo para verificar tanto la integridad como la autenticidad de un mensaje.
- **HSM (Hardware Security Module):** Módulo de seguridad de hardware; dispositivo criptográfico físico que protege y gestiona claves digitales.
- **HSS/HLR (Home Subscriber Server/Home Location Register):** Servidor de abonados de origen / Registro de ubicación de origen; base de datos central en redes móviles que contiene la información de los suscriptores.
- **HTTP (Hypertext Transfer Protocol):** Protocolo de transferencia de hipertexto, utilizado para la comunicación en la World Wide Web.
- **HTTPS (HTTP Secure):** Versión segura de HTTP que utiliza TLS/SSL para cifrar la comunicación.
- **ICCID (Integrated Circuit Card Identifier):** Identificador de tarjeta de circuito integrado; número único que identifica a cada tarjeta SIM o perfil eSIM.
- **IETF (Internet Engineering Task Force):** Grupo de trabajo de ingeniería de Internet; organización que desarrolla y promueve estándares de Internet, incluyendo protocolos como TLS y CoAP.
- **IMSI (International Mobile Subscriber Identity):** Identidad Internacional del abonado móvil; número único que identifica a un usuario en una red móvil.
- **IoT (Internet of Things):** Internet de las cosas; red de dispositivos físicos interconectados que recogen e intercambian datos.
- **IPA (IoT Profile Assistant):** Asistente de perfil para IoT; componente de software (que puede residir en el dispositivo o en la eUICC) encargado de gestionar los perfiles eSIM.
- **IPAd (IoT Profile Assistant on device):** Variante del IPA que reside directamente en el software del dispositivo IoT.

- **IPAe (IoT Profile Assistant on eUICC):** Variante del IPA que está integrada dentro del chip eUICC.
- **IPFS (InterPlanetary File System):** Sistema de archivos Interplanetario; un protocolo y red peer-to-peer para almacenar y compartir datos de forma distribuida.
- **ISD-P (Issuer Security Domain - Profile):** Dominio de seguridad del emisor - perfil; un área segura dentro de la eUICC que contiene y protege un perfil de operador específico.
- **ISD-R (Issuer Security Domain - Root):** Dominio de Seguridad del Emisor - Raíz; el dominio de seguridad principal en una eUICC, que gestiona la instalación y el ciclo de vida de todos los perfiles (ISD-Ps).
- **JSON (JavaScript Object Notation):** Formato de texto ligero para el intercambio de datos, fácil de leer para los humanos y de procesar para las máquinas.
- **JSON-RPC (JSON Remote Procedure Call):** Protocolo de llamada a procedimiento remoto que utiliza JSON para codificar los mensajes.
- **JWT (JSON Web Token):** Estándar abierto para crear tokens de acceso que permiten la propagación segura de identidades y permisos entre partes.
- **LPWAN (Low-Power Wide-Area Network):** Red de área amplia y baja potencia; tipo de red de telecomunicaciones diseñada para comunicaciones de largo alcance con bajo consumo de energía, ideal para IoT.
- **MNO (Mobile Network Operator):** Operador de Red Móvil; empresa que provee servicios de comunicación inalámbrica a suscriptores de telefonía móvil.
- **MQTT (Message Queuing Telemetry Transport):** Protocolo de mensajería ligero basado en el modelo publicación/suscripción, optimizado para dispositivos con recursos limitados.
- **NB-IoT (Narrowband IoT):** Estándar de radio LPWAN para conectar una amplia gama de dispositivos y servicios IoT.
- **NCD (No-User-Interface Constrained Device):** Dispositivo con restricciones y sin interfaz de usuario, típico en el entorno IoT.
- **NIST (National Institute of Standards and Technology):** Instituto Nacional de Estándares y Tecnología de EE. UU., que publica guías y recomendaciones sobre criptografía y seguridad.
- **NoSQL:** Tipo de base de datos que no utiliza el lenguaje SQL tradicional y está diseñada para modelos de datos flexibles y escalables.

- **OCSP (Online Certificate Status Protocol):** Protocolo de Internet utilizado para obtener el estado de revocación de un certificado digital X.509.
- **OSS (Operations Support System):** Sistema de soporte a las operaciones; conjunto de programas que ayudan a un proveedor de servicios a monitorizar, controlar y gestionar una red de telecomunicaciones.
- **OTA (Over-the-Air):** Método de distribución de actualizaciones de software o configuración de forma remota a través de redes inalámbricas.
- **PFS (Perfect Forward Secrecy):** Propiedad de los protocolos de acuerdo de clave que asegura que el compromiso de claves a largo plazo no compromete la seguridad de las claves de sesión pasadas.
- **PKI (Public Key Infrastructure)**: Infraestructura de Clave Pública; sistema para crear, gestionar y distribuir certificados digitales y claves públicas.
- **PoC (Proof of Concept):** Prueba de Concepto; una demostración para verificar que una idea o teoría tiene potencial práctico.
- **QoS (Quality of Service):** Calidad de Servicio; medición del rendimiento general de un servicio, como una red de telecomunicaciones, percibido por los usuarios.
- **RADIUS (Remote Authentication Dial-In User Service):** Protocolo de red que proporciona autenticación, autorización y contabilidad centralizadas para usuarios que se conectan y utilizan un servicio de red.
- **REST (Representational State Transfer):** Arquitectura de software para sistemas distribuidos como la World Wide Web, comúnmente utilizada para crear APIs.
- **RFC (Request for Comments):** Publicación de la IETF que documenta estándares, protocolos y mejores prácticas de Internet.
- **RSP (Remote SIM Provisioning):** Aprovisionamiento remoto de SIM; la tecnología y el proceso estandarizado por la GSMA para gestionar perfiles de eSIM de forma remota.
- **SDN (Software-Defined Networking):** Redes definidas por software; un enfoque de la arquitectura de red que permite que la red sea controlada de forma centralizada mediante software.
- **SIM (Subscriber Identity Module):** Módulo de identidad del abonado; la tarjeta tradicional que identifica a un usuario en una red móvil.
- **SM-DP+ (Subscription Manager - Data Preparation+):** Gestor de suscripciones - Preparación de datos; componente del ecosistema eSIM responsable de la creación, gestión y entrega segura de perfiles de operador.
- **SM-DS (Subscription Manager - Discovery Server):** Gestor de suscripciones - Servidor de descubrimiento; componente que permite a un dispositivo descubrir si hay perfiles eSIM disponibles para él.

- **SM-SR (Subscription Manager - Secure Routing):** Gestor de suscripciones - Enrutamiento seguro; componente que en arquitecturas más antiguas gestionaba el enrutamiento seguro de los comandos de gestión de perfiles.
- **SPTP (SIM Profile Transparency Protocol):** Protocolo de transparencia de perfiles SIM; una propuesta de protocolo para detectar la provisión maliciosa de perfiles mediante un registro auditable.
- **SQL (Structured Query Language):** Lenguaje de consulta estructurado, utilizado para gestionar y consultar datos en bases de datos relacionales.
- **TEE (Trusted Execution Environment):** Entorno de ejecución confiable; un área segura dentro del procesador principal de un dispositivo que garantiza que el código y los datos cargados allí estén protegidos.
- **TLS (Transport Layer Security):** Seguridad de la Capa de Transporte; protocolo criptográfico que proporciona seguridad a las comunicaciones sobre una red de computadoras.
- **TLV (Tag-Length-Value):** Etiqueta-Longitud-Valor; un formato de codificación de datos donde cada elemento de información está precedido por un identificador (etiqueta) y su longitud.
- **TPM (Trusted Platform Module):** Módulo de Plataforma Confiable; un estándar internacional para un criptoprocesador seguro, un microcontrolador que puede almacenar artefactos criptográficos.
- **UICC (Universal Integrated Circuit Card):** Tarjeta de circuito integrado universal; la plataforma de tarjeta inteligente segura que contiene una aplicación SIM. La eSIM es una forma de UICC.
- **UI (User Interface):** Interfaz de Usuario.
- **UML (Unified Modeling Language):** Lenguaje de modelado unificado; lenguaje estandarizado para la visualización, especificación, construcción y documentación de sistemas de software.
- **UDP (User Datagram Protocol):** Protocolo de datagramas de usuario, un protocolo de comunicación sin conexión que no garantiza la entrega de paquetes.
- **URL (Uniform Resource Locator):** Localizador Uniforme de Recursos; la dirección de un recurso en la World Wide Web.
- **VRF (Verifiable Random Function):** Función aleatoria verificable; una función criptográfica que produce una salida pseudoaleatoria y demostrable, utilizada en el protocolo SPTP.
- **WSGI (Web Server Gateway Interface):** Interfaz de pasarela de servidor web; una especificación simple para que los servidores web pasen peticiones a aplicaciones web o frameworks escritos en Python.

- **WSL (Windows Subsystem for Linux):** Subsistema de Windows para Linux; una capa de compatibilidad para ejecutar ejecutables de Linux de forma nativa en Windows.
- **XCM (Cross-Chain Messaging):** Mensajería entre Cadenas; formato para la comunicación entre diferentes parachains en el ecosistema Polkadot.

Índice de tablas

Capítulo 1 - Introducción

En la era digital actual, la conectividad y la seguridad de los dispositivos IoT (Internet de las Cosas) son pilares fundamentales para el desarrollo de soluciones inteligentes y sostenibles. Uno de los avances más significativos en este ámbito ha sido la evolución de la tradicional tarjeta SIM, una tarjeta física que permite conectar dispositivos a redes móviles, hacia la eSIM (embedded SIM), un chip integrado directamente en el hardware del dispositivo. Esta transición no solo elimina la necesidad de insertar o reemplazar tarjetas físicas, sino que permite descargar y gestionar múltiples perfiles de operadores de forma remota y segura. Gracias a esta innovación, se ha ganado en flexibilidad, eficiencia y escalabilidad para entornos con gran cantidad de dispositivos conectados. No obstante, esta evolución también introduce nuevos retos en cuanto a la seguridad de los datos, la interoperabilidad entre plataformas y la administración confiable de identidades digitales.

La tecnología blockchain, caracterizada por su naturaleza descentralizada, con historial inmutable y transparente, podría posicionarse como una solución innovadora para abordar estos desafíos. Su aplicación en el ecosistema de eSIMs en dispositivos IoT permite mejorar la seguridad de las transacciones de datos, garantizar la autenticidad de los dispositivos y proporcionar un mecanismo confiable para la gestión de identidades digitales. Además, blockchain facilita la interoperabilidad entre diferentes proveedores y plataformas, permitiendo un entorno más eficiente y seguro, sin dependencia de un proveedor central.

Este trabajo aborda la aplicación de la tecnología blockchain en la securización e interoperabilidad de eSIMs en dispositivos IoT. Se analizarán los beneficios, desafíos y casos de uso más relevantes, con el objetivo de proporcionar un marco conceptual que facilite la adopción de estas tecnologías en entornos industriales y comerciales. A través de este análisis, se busca demostrar cómo la combinación de blockchain y eSIMs puede revolucionar la conectividad y la seguridad en el creciente mundo del IoT.

1.1. Motivación

Los entornos IoT están en constante expansión, con un número elevado de dispositivos conectados en diversos sectores como ciudades inteligentes, vehículos autónomos, gestión de activos, telemedición, geolocalización, agricultura, finanzas y salud. Sin embargo, la escalabilidad y la interoperabilidad de estas soluciones siguen siendo grandes desafíos, especialmente en lo que respecta a la gestión de eSIMs y la conectividad entre múltiples operadores de red.

Actualmente, no existe un mecanismo eficiente y descentralizado que permita la gestión transparente de eSIMs entre distintos operadores de red móvil sin depender de plataformas centralizadas. Esta fragmentación complica la administración de perfiles de eSIM y dificulta la migración entre proveedores, lo que genera ineficiencias y altos costos operativos. Asimismo, la falta de trazabilidad sobre el historial de conexión y operación de los dispositivos IoT limita la capacidad de monitoreo y auditoría, afectando tanto la seguridad como la confiabilidad de las redes.

La descentralización de la gestión de eSIMs mediante blockchain permitiría la creación de un registro distribuido y auditable, eliminando la dependencia de plataformas específicas de operadores y facilitando la interoperabilidad. Un sistema basado en blockchain permitiría que diferentes operadores accedan a un mismo entorno de gestión, optimizando la administración de perfiles de eSIMs sin depender de un solo proveedor.

La trazabilidad y auditoría de dispositivos IoT es otro aspecto que puede mejorarse con blockchain. Su capacidad para almacenar registros inmutables permitiría un seguimiento detallado del historial de conectividad y eventos operacionales de cada dispositivo IoT. Esto no solo mejoraría la seguridad, sino que también facilitaría la detección de anomalías y posibles intentos de fraude o uso indebido de los dispositivos.

En términos de seguridad y autenticación, blockchain puede proporcionar mecanismos robustos que reduzcan los riesgos de fraude o suplantación mediante el uso de identidades digitales verificables y contratos inteligentes. Cada dispositivo podría tener una identidad única y verificable en la cadena de bloques, lo que garantiza que sólo los dispositivos autorizados puedan operar dentro de la red IoT.

Este trabajo tiene como objetivo evaluar la viabilidad de estas soluciones y explorar cómo blockchain puede transformar la forma en que se gestionan las

eSIMs y la conectividad en dispositivos IoT, impulsando un ecosistema más seguro, eficiente y transparente.

Es importante subrayar que la tecnología blockchain no es una solución universal. Somos conscientes de que su implementación en un entorno productivo presenta desafíos significativos en cuanto a costes operativos, la definición de un modelo de gobernanza claro entre los actores y el necesario encaje en el marco regulatorio vigente. Además, la tecnología en sí no está exenta de retos de usabilidad [74] y su aplicación no siempre es la más apropiada, por lo que su elección debe estar bien justificada [75].

En este contexto, la presente investigación explora la tecnología blockchain como una alternativa viable para abordar las limitaciones de los modelos centralizados en la gestión de eSIMs. Mientras que una base de datos tradicional controlada por un único actor perpetúa la dependencia y la falta de confianza entre competidores, las características fundamentales de una blockchain como la capacidad de crear un registro inmutable y auditable compartido entre todas las partes presentan una oportunidad para diseñar un sistema con mayor transparencia y resiliencia.

Por lo tanto, el enfoque de este trabajo no pretende postular la blockchain como la única solución, sino estudiar su viabilidad como una herramienta estratégica para construir un ecosistema de gestión de eSIMs más abierto, interoperable y confiable, que es precisamente el objetivo de esta investigación.

1.2. Objetivos

El objetivo principal de este trabajo es diseñar un modelo basado en blockchain que resuelva los problemas identificados en la gestión de eSIMs en dispositivos IoT, optimizando la seguridad, interoperabilidad y trazabilidad sin depender de plataformas centralizadas. Para garantizar que esta solución sea viable y alineada con estándares industriales, se explorará la integración con los estándares de la GSMA (Global System for Mobile Communications Association) con el fin de evaluar cómo blockchain puede complementarlos o incluso reemplazar ciertas funciones para mejorar su eficiencia y seguridad.

Entre los objetivos específicos de este trabajo se encuentran:

- Desarrollar un **modelo de gestión descentralizada de eSIMs** basado en blockchain que permita a un dispositivo IoT conectarse a múltiples

operadores de manera eficiente sin depender de infraestructuras propietarias.

- Implementar un **mecanismo de trazabilidad** que registre de forma inmutable el historial de conexión y operación de cada dispositivo IoT, permitiendo auditorías y análisis de seguridad más efectivos.
- Diseñar un sistema de **autenticación** basado en blockchain que garantice la identidad única y verificable de cada dispositivo IoT, reduciendo el riesgo de suplantación o acceso no autorizado.
- Explorar cómo blockchain puede integrarse con los **estándares de la GSMA** (SGP.31, SGP.32,) para mejorar la seguridad e interoperabilidad de eSIMs en IoT o, en su defecto, evaluar un modelo alternativo que reemplace alguna de sus funciones.

Este enfoque busca desarrollar una solución innovadora y viable, con fundamentos técnicos sólidos y factibilidad de implementación en entornos reales. La propuesta no solo abordará los desafíos actuales, sino que también ofrecerá mejoras tangibles en la seguridad y la interoperabilidad del ecosistema IoT, garantizando que el modelo propuesto sea compatible con estándares industriales y normativas globales.

1.3. Plan de trabajo

El desarrollo de este trabajo se llevó a cabo siguiendo un plan estructurado en cinco fases claves, ejecutadas a lo largo de un periodo de seis meses:

1. **Fase de investigación y fundamentación:** Se realizó una investigación exhaustiva de los fundamentos teóricos, incluyendo la tecnología blockchain, el ecosistema eSIM para IoT y los estándares de la GSMA (principalmente SGP.31 y SGP.32). El objetivo de esta fase fue consolidar la base de conocimiento necesaria para el proyecto.

2. **Fase de análisis y definición del problema:** Se analizaron las limitaciones de los modelos centralizados actuales en la gestión de eSIMs para identificar los desafíos claves en seguridad, interoperabilidad y trazabilidad. En esta etapa se definió el valor diferencial y el alcance de la solución propuesta, centrada en la gestión de identidades, la autenticación de dispositivos y la provisión descentralizada de perfiles.

3. **Fase de diseño de la arquitectura:** Se diseñaron las dos arquitecturas técnicas que constituyen el núcleo de este trabajo. El diseño incluyó la definición de los componentes (IPA, backend, eIM), los flujos de interoperabilidad entre los actores del ecosistema (dispositivo, proveedor de servicios, MNO) y la lógica de los contratos inteligentes para gestionar los eventos en la blockchain.

4. **Fase de desarrollo y validación técnica:** Se implementó una prueba de concepto (PoC) para validar la viabilidad técnica de la arquitectura. El prototipo se centró en demostrar la interacción entre el backend, la base de datos y los contratos inteligentes para orquestar operaciones como el registro de dispositivos y el cambio de operador, tal como se detalla en el Capítulo 6.

5. **Fase de evaluación y conclusiones:** Finalmente, se evaluó la propuesta desde una doble perspectiva: se analizaron los resultados técnicos de la PoC y, en paralelo, se realizó una validación cualitativa mediante una encuesta a expertos del sector. Esto permitió extraer conclusiones sobre la viabilidad y los desafíos del modelo, así como redactar la memoria final de este trabajo.

1.4. Estructura del documento

El presente documento está compuesto de 7 capítulos que describen el uso de la Tecnología Blockchain aplicada a la securización e interoperabilidad de eSIMs en dispositivos IoT. A continuación, se presenta un breve resumen del contenido de cada uno de estos capítulos.

Capítulo 1: Introducción

Este capítulo presenta el contexto y la motivación detrás de la investigación, resaltando la importancia de la seguridad y la interoperabilidad en el ecosistema IoT con eSIMs. Se describen los objetivos del trabajo y se expone el plan de trabajo que guiará el desarrollo del estudio.

Capítulo 2: Fundamentos teóricos

Esta sección presenta los fundamentos esenciales para entender la investigación, incluyendo los principios de la tecnología blockchain, sus tipos, características, y el uso de smart contracts. Se abordan también los conceptos clave del IoT y de la tecnología eSIM, junto con los estándares GSMA SGP.31 y

SGP.32. En el estado del arte se analizan enfoques que integran blockchain, eSIM y protocolos seguros para mejorar la gestión y autenticación de dispositivos IoT. Se destacan soluciones orientadas a la automatización del aprovisionamiento, la autenticación descentralizada y la protección de credenciales, incluyendo el estándar IoT SAFE. Estas propuestas sirven como base técnica para la solución planteada en este trabajo.

Capítulo 3: Metodologías y tecnologías

Este capítulo presenta el enfoque metodológico adoptado para el desarrollo del trabajo, estructurado en torno a tres dimensiones: investigación documental, desarrollo experimental y validación técnica. Se describen las fuentes utilizadas para el análisis normativo y técnico, así como las herramientas y lenguajes de programación empleados en la construcción del prototipo. También se explican los criterios para la aplicación de la encuesta a expertos, el modelo de muestreo utilizado y el rol de las tutorías académicas. Finalmente, se detalla el conjunto de tecnologías y plataformas utilizadas tanto en el diseño como en la implementación del sistema propuesto.

Capítulo 4: Arquitectura técnica propuesta

Este capítulo analiza cómo la tecnología blockchain puede mejorar la gestión de eSIMs en dispositivos IoT. Se examinan modelos existentes y se proponen dos arquitecturas basadas en blockchain que permite la gestión remota y segura de perfiles eSIM, optimizando la interoperabilidad entre operadores móviles. Se presentan además casos de uso, ventajas y desafíos de la solución planteada.

Capítulo 5: Validación de la propuesta

Este capítulo presenta la validación de las arquitecturas propuestas mediante dos enfoques complementarios. Primero, se documenta la prueba de concepto (PoC) técnica desarrollada, que demuestra la viabilidad de la interacción entre el backend y la blockchain para orquestar operaciones claves como el registro de dispositivos y el cambio de operador. Segundo, se presentan y analizan los resultados de una encuesta dirigida a profesionales del sector para evaluar la viabilidad, el valor y el grado de innovación de la solución, aportando así una validación tanto empírica como opinión experta a la investigación.

Capítulo 6: Conclusiones y trabajo futuro

En esta sección se resumen los hallazgos claves del trabajo, destacando las mejoras en seguridad y eficiencia que ofrece blockchain en la gestión de eSIMs. Se identifican también los desafíos aún pendientes y se proponen líneas de investigación futura para continuar con el desarrollo de soluciones innovadoras en este ámbito.

Capítulo 2 - Fundamentos Teóricos

Con el propósito de abordar los retos que surgen de los ecosistemas de IoT, y en particular en la gestión segura e interoperable de perfiles eSIM en dispositivos IoT, es necesario comprender las tecnologías que sustentan esta propuesta. En este capítulo se explican los fundamentos teóricos de cuatro componentes tecnológicos, Blockchain, IoT, eSIM y los estándares de la GSMA, no solo como componentes por sí solos, sino en función del rol de cada uno de ellos dentro de la arquitectura que aquí se planteará. El objetivo no es únicamente describir su funcionamiento, sino también sentar las bases que justifiquen su integración como una solución global.

2.1. La tecnología blockchain

La cadena de bloques, es una tecnología de registro distribuido que permite el almacenamiento y gestión de información de manera segura, historial inmutable y descentralizado. Consiste en una base de datos compartida entre todos los participantes de una red, eliminando la necesidad de una autoridad central para su control. Los datos se almacenan en bloques que están interconectados mediante hashes criptográficos, formando una cadena. Cada bloque incluye el hash del bloque anterior, lo que asegura la integridad de la cadena. Además, para añadir un nuevo bloque, se requiere la aprobación de los participantes de la red, lo que garantiza la transparencia y la confianza en el sistema [1][2].

Esta tecnología, ampliamente conocida por ser la base de las criptomonedas como Bitcoin, se ha expandido a otros usos, como el seguimiento de activos, la gestión de contratos inteligentes y la optimización de cadenas de suministro. Su resistencia a manipulaciones se basa en principios criptográficos y modelos de consenso, como la prueba de trabajo (Proof of Work) o la prueba de participación (Proof of Stake) [3][4]. Sin embargo, enfrenta retos como la escalabilidad y el consumo energético, particularmente en redes públicas.

2.1.1. Tipos

Con el paso del tiempo blockchain ha podido adaptarse a diversas necesidades y aplicaciones, lo que ha dado lugar a la categorización de diferentes tipos de blockchains según su estructura y control. Aunque todas comparten la

característica de ser registros distribuidos e inmutables, difieren en términos de quién puede participar en la red, validar transacciones y acceder a la información almacenada. Estas diferencias determinan su idoneidad para distintos casos de uso [56], desde redes completamente abiertas hasta sistemas restringidos y controlados. Entre las principales clasificaciones se encuentran las blockchains públicas, privadas, híbridas y de consorcio, cada una con características únicas que las hacen aptas para contextos específicos. Este enfoque permite a las organizaciones y comunidades seleccionar el modelo que mejor se adapte a sus objetivos, equilibrando transparencia, seguridad y control.

2.1.1.1. Blockchain pública

Una blockchain pública permite a cualquier persona unirse, participar y verificar las transacciones sin necesidad de permisos. Todas las transacciones realizadas en una blockchain pública son transparentes y visibles para todos los participantes, lo que garantiza un alto nivel de confianza y seguridad. Este tipo de blockchain es ideal para aplicaciones que requieren transparencia total, como las criptomonedas, incluyendo Bitcoin y Ethereum [1][4].

2.1.1.2. Blockchain privada

En las blockchains privadas, una organización centralizada controla la red, y el acceso está restringido únicamente a los participantes autorizados. Estas redes resultan útiles para entornos empresariales que necesitan un alto grado de privacidad y control sobre los datos compartidos, como la gestión de cadenas de suministro o transacciones financieras internas [5]. Aunque estas redes son más eficientes que las blockchains públicas en términos de rendimiento, su estructura centralizada puede hacerlas vulnerables a manipulaciones internas y levantar preocupaciones sobre la confianza entre los participantes [6][7].

2.1.1.3. Blockchain de consorcio

También conocidas como blockchains federadas, estas redes son gestionadas conjuntamente por un grupo de organizaciones que colaboran para mantener el registro distribuido. A diferencia de las blockchains públicas, el acceso está limitado a los miembros del consorcio, lo que proporciona un equilibrio entre transparencia y privacidad [8]. Este modelo es ampliamente utilizado en sectores como el financiero y la logística, donde múltiples entidades necesitan interactuar de manera segura y eficiente [9][10]. Aunque comparten características con las blockchains privadas, las de consorcio no están controladas por una sola entidad,

sino por un conjunto de actores predefinidos, lo que las hace más resilientes a conflictos de intereses internos [6].

2.1.1.4. Blockchain híbrida

Las blockchains híbridas combinan características de las blockchains públicas y privadas, permitiendo un acceso parcialmente controlado. Esto significa que ciertas transacciones o datos son accesibles para el público, mientras que otras están restringidas a usuarios autorizados. Este tipo de blockchain es ideal para aplicaciones que requieren flexibilidad en términos de transparencia y control, como la gestión de identidades o contratos inteligentes en entornos gubernamentales [2][11]. Esta combinación permite a las organizaciones implementar soluciones personalizables que aprovechan las ventajas de ambas arquitecturas mientras mitigan sus desventajas [12].

2.1.2. Smart Contracts

Los smart contracts son programas informáticos que ejecutan automáticamente los términos y condiciones definidos en un contrato, eliminando la necesidad de intermediarios o terceros de confianza entre las partes [13]. Propuestos originalmente por Nick Szabo en la década de 1990, los smart contracts buscan replicar los acuerdos legales en un formato digital, ofreciendo mayor eficiencia y seguridad [14].

En esencia, un smart contract es un código almacenado en una blockchain que define reglas y consecuencias, las cuales se ejecutan automáticamente cuando se cumplen ciertas condiciones predefinidas. Esto significa que, una vez desplegado en la red, su ejecución es irreversible y transparente para todos los participantes del sistema [15].

Los smart contracts tienen un amplio rango de aplicaciones, desde la automatización de pagos hasta la ejecución de contratos financieros complejos, la gestión de activos digitales, y el control de procesos en cadenas de suministro [16]. Gracias a la infraestructura blockchain, los smart contracts permiten transacciones entre partes no confiables sin la necesidad de recurrir a intermediarios, reduciendo costos y riesgos [17].

Sin embargo, a pesar de sus ventajas, presentan desafíos técnicos y legales [57]. Por un lado, su seguridad depende de la precisión y robustez del código, ya que errores o vulnerabilidades pueden ser explotados, como ocurrió en el famoso

caso del hackeo de "The DAO" en 2016 [18][58]. Por otro lado, la falta de regulación clara en muchas jurisdicciones plantea preguntas sobre su validez legal y responsabilidad en caso de fallos [19].

Finalmente, los smart contracts son una tecnología disruptiva que está transformando sectores como el financiero y el logístico. Gracias a su capacidad para reducir costos operativos, aumentar la transparencia y garantizar la ejecución automática de acuerdos, se proyectan como una pieza clave en el futuro de los sistemas descentralizados [20].

2.1.3 Plataformas

Las plataformas de blockchain son infraestructuras digitales que permiten la creación y gestión de redes descentralizadas. Ofreciendo un entorno seguro y transparente para la ejecución de transacciones y contratos inteligentes. Estas plataformas han revolucionado diversos sectores al proporcionar mecanismos confiables para el intercambio de información y valor sin la necesidad de intermediarios tradicionales.

Aunque existen múltiples plataformas como Hyperledger Fabric, orientada a entornos empresariales o Polkadot, enfocada en la interoperabilidad entre cadenas, Ethereum se ha consolidado como la plataforma de referencia para el desarrollo de aplicaciones descentralizadas (dApps).

Propuesta originalmente en 2014 por Vitalik Buterin, Ethereum fue pionera al introducir el concepto de contratos inteligentes (smart contracts) de propósito general. A diferencia de Bitcoin, que fue diseñado principalmente como un sistema de dinero electrónico. Ethereum fue diseñado para funcionar como una red global descentralizada que permite ejecutar y verificar programas de forma distribuida." [14].

El núcleo de esta capacidad es la Máquina Virtual de Ethereum (EVM), un entorno de ejecución completo que permite ejecutar cualquier script algorítmicamente complejo, siempre que se disponga de los recursos necesarios. Los contratos inteligentes, comúnmente escritos en lenguajes de alto nivel como Solidity, se compilan a bytecode que la EVM puede interpretar y ejecutar.

Para evitar el abuso de la red y compensar el coste computacional, Ethereum introdujo el concepto de gas. Cada operación en la EVM tiene un coste fijo en unidades de gas, por lo que las transacciones y la ejecución de contratos

requieren una comisión que se paga en la criptomoneda nativa de la red, Ether (ETH). Este mecanismo no solo previene bucles infinitos y ataques de denegación de servicio, sino que también incentiva la escritura de código eficiente.

Originalmente, Ethereum utilizaba un mecanismo de consenso de Prueba de Trabajo (Proof-of-Work), pero ha evolucionado hacia un modelo de Prueba de Participación (Proof-of-Stake), mucho más eficiente energéticamente. Gracias a su madurez, su amplia comunidad de desarrolladores y su robusta infraestructura, Ethereum es la base de la mayoría de aplicaciones en finanzas descentralizadas (DeFi), tokens no fungibles (NFTs) y, como se explora en este trabajo, soluciones para el Internet de las Cosas.

Otra plataforma relevante es Hyperledger Fabric, como bien lo mencionamos al iniciar esta sección. Impulsada por la Fundación Linux, que se enfoca en soluciones empresariales. A diferencia de las blockchains públicas, Hyperledger Fabric ofrece una arquitectura modular y permisos de acceso controlados, lo que la hace ideal para aplicaciones que requieren privacidad y rendimiento en entornos corporativos [8].

Además, plataformas como Corda se han especializado en el sector financiero, proporcionando una infraestructura diseñada para registrar, gestionar y sincronizar acuerdos financieros entre instituciones de manera eficiente y segura [21].

Estas plataformas varían en sus enfoques y arquitecturas, pero comparten el objetivo común de ofrecer soluciones descentralizadas que mejoran la seguridad y eficiencia en diversas industrias. La elección de una plataforma específica depende de las necesidades particulares de cada aplicación, considerando factores como el modelo de consenso, la escalabilidad y los requisitos de privacidad.

A continuación, se muestra una tabla 2.1 comparativa entre las principales plataformas de blockchain:

Característica	Ethereum	Hyperledger Fabric	Algorand	Avalanche	Polkadot
Tipo de Blockchain	Pública	Privada Consorcio	Pública	Pública con subredes	Pública con Relay Chain
Modelo de Consenso	PoW → PoS (Ethereum 2.0)	PBFT o RAFT (modular)	Pure Proof-of-Stake (PPoS)	Avalanche Consensus (Snowman + DAG)	NPoS + GRANDPA
Lenguaje de Programación	Solidity	Go, Java, JavaScript	TEAL (Algorand Smart Contract Language)	Solidity y EVM-compatible	Rust, Ink!, otros
Contratos Inteligentes	Sí (EVM)	Chaincode en múltiples lenguajes	Smart Contracts en TEAL y PyTEAL	Smart Contracts en Solidity o Vyper	Compatible con EVM y otros lenguajes
Mecanismo de Ejecución	EVM	Docker + Chaincode	Máquina de contratos TEAL	EVM (C-Chain)	EVM y contratos nativos
Tiempo de Confirmación	12-15s (PoS)	<1s (red optimizada)	4-5s aprox.	Subsegundos (alta escalabilidad)	6-20s (con GRANDPA)
Costo de Transacción	Gas variable (congestión)	Sin costos directos	Bajo (fracciones de centavo)	Muy bajo	Bajo, depende de la parachain
Escalabilidad	Limitada	Alta mediante arquitectura modular	Alta (1000+ TPS)	Alta (subredes paralelas)	Alta (parachains paralelas)
Interoperabilidad	Sidechains	Múltiples blockchains vía canales	Limitada, en desarrollo	Alta vía subredes interoperables	Nativa vía XCM (Cross-Chain Messaging)
Gobernanza	Descentralizada	Fundaciones o consorcios	Gobernanza algorítmica	Gobernanza on-chain con staking	Gobernanza on-chain y referendos
Seguridad y Descentralización	Alta, pero sujeta a congestión	Alta en entorno cerrado	Alta: validadores seleccionados y rotación	Alta tolerancia a fallos, descentralizado	Alta, seguridad compartida de Relay Chain
Latencia y Finalidad	Finalidad probabilística	Baja latencia, rápida finalidad	4-5s	Finalidad en <1s	Finalidad en 6-20s aprox.
Ecosistema IoT y Aplicaciones	DeFi, NFTs, dApps	Supply chain, salud, identidad	Proyectos iniciales IoT y tokenización de activos	IoT crítico, interoperabilidad, DeFi	Aplicaciones IoT, conectividad entre blockchains
Comunidad y Desarrollo	Amplia y madura	Empresarial y académica	En crecimiento, fuerte en DeFi	Activa y orientada a alto rendimiento	Muy activa, diversidad de proyectos

Tabla 2.1. Comparativa de plataformas Blockchain

Una vez comprendidas las propiedades fundamentales de la tecnología blockchain como la inmutabilidad, descentralización y trazabilidad y sus capacidades mediante smart contracts, resulta conveniente examinar el entorno sobre el cual se aplicarán estas ventajas. En este sentido, el ecosistema de dispositivos IoT representa un entorno desafiante, donde millones de dispositivos con capacidades heterogéneas, requieren conectividad segura, automatizada y auditable. A continuación, se profundiza en las características del IoT, su constante crecimiento y las particularidades que lo convierten en un caso de uso sensible a problemas de identidad, confianza y gestión en la operación, que precisamente blockchain puede llegar a resolver.

2.2. Internet de las cosas IoT

El Internet de las Cosas (IoT) se puede definir como una red interconectada de dispositivos físicos como sensores, actuadores, electrodomésticos y otros objetos embebidos con hardware y software que les permite recoger, intercambiar y procesar datos en tiempo real. La combinación entre infraestructura física y digital, facilita la automatización y el monitoreo de procesos en múltiples sectores [24]. En este sentido, IoT no solo se refiere a la conectividad, sino también a la capacidad de análisis y toma de decisiones basada en datos provenientes de diversos orígenes.

Dentro del ecosistema IoT, la seguridad y la interoperabilidad son dos pilares fundamentales. Asegurar un ecosistema IoT implica proteger los datos frente a accesos no autorizados, manipulaciones y fallos de disponibilidad, pilares esenciales de la ciberseguridad. Roman et al. [25] destacan que, debido a la heterogeneidad de dispositivos y protocolos, la protección contra ciberataques y la gestión segura de la identidad son desafíos críticos en este ámbito. Por otro lado, la interoperabilidad se refiere a la capacidad de distintos dispositivos, redes y plataformas de trabajar conjuntamente mediante estándares comunes, lo que es esencial para la integración eficiente de sistemas y para el correcto funcionamiento de aplicaciones complejas [26].

A mediados de esta década, el ecosistema IoT muestra una madurez evidente gracias a la expansión de tecnologías como 5G y el avance hacia redes 6G y computación en la nube. La integración de plataformas de blockchain en IoT se ha consolidado como una solución efectiva para abordar desafíos de seguridad e interoperabilidad. Integrar blockchain en soluciones IoT ha permitido delegar la

gestión de identidades y transacciones en estructuras descentralizadas, aumentando su resistencia frente a amenazas. [27].

La versión de eSIM diseñada para IoT responde a necesidades específicas como bajo consumo, gestión remota y ausencia de interfaz física basada en especificaciones adicionales (como la GSMA SGP.31) que tienen en cuenta las restricciones de energía, conectividad y la necesidad de provisión remota en dispositivos sin interfaz de usuario tradicional. Esta evolución permite la activación y gestión de perfiles de operador de manera dinámica, lo que es crucial en entornos donde intervenciones físicas son limitadas o imposibles [23]. Además, la combinación de blockchain con eSIM IoT crea un marco robusto en el que cada transacción como la activación, cambio de perfil o actualización de credenciales queda registrada de manera inmutable, facilitando la auditoría y el cumplimiento normativo.

La interoperabilidad se ha convertido en un requisito indispensable en el contexto IoT actual. Los estándares desarrollados por organizaciones como la GSMA y la Asociación de Tercera Generación (3GPP)[55] aseguran que las soluciones de eSIM sean compatibles a nivel global, permitiendo la integración de dispositivos y servicios de diferentes proveedores y facilitando la migración y actualización de infraestructuras sin interrupciones significativas [28].

Uno de los avances más relevantes para facilitar la conectividad flexible y remota en IoT ha sido la evolución de las tarjetas SIM tradicionales hacia soluciones embebidas, como las eSIM. Este cambio tecnológico permite el aprovisionamiento remoto de perfiles de red y elimina la necesidad de intervención física en cada dispositivo desplegado. Sin embargo, también abre nuevas puertas a riesgos asociados a la integridad de los procesos de activación, portabilidad o auditoría. A continuación, se analiza en detalle el funcionamiento del sistema eSIM, sus actores y su ciclo de vida, sentando las bases para comprender por qué es necesario reforzar su infraestructura con tecnologías como blockchain.

2.3. Tecnología eSIM IoT

La eSIM (Embedded Subscriber Identity Module) representa una evolución significativa en la tecnología de identificación de abonados en redes móviles. A diferencia de las tarjetas SIM tradicionales, la eSIM está integrada directamente en el dispositivo, eliminando la necesidad de una tarjeta física extraíble. Esta

innovación ha sido estandarizada por organizaciones como la GSMA y el 3GPP, que han desarrollado especificaciones detalladas para su implementación y funcionamiento.

Definición Formal:

Según la GSMA, una eSIM es una UICC (Universal Integrated Circuit Card) programable que está embebida en un dispositivo y que puede gestionar múltiples perfiles de operador, permitiendo la provisión remota de estos perfiles sin necesidad de intervención física. Esta definición se encuentra en la especificación SGP.21 de la GSMA, que detalla la arquitectura y los requisitos para la eSIM en dispositivos de consumo [22].

Estándares y Protocolos:

La GSMA ha publicado varias especificaciones clave para la eSIM, entre las que destacan:

- **SGP.21**: Especificación de Arquitectura de eSIM para el Consumidor, que define la arquitectura general y los componentes involucrados en la implementación del eSIM en dispositivos de consumo [22].
- **SGP.22**: Especificación de Requisitos de eSIM para el Consumidor, que establece los requisitos técnicos y de seguridad que deben cumplir los dispositivos y las plataformas de gestión de eSIM.

A medida que la tecnología evoluciona, la implementación de la eSIM se ha extendido al ámbito del Internet de las Cosas (IoT), dando lugar a la eSIM IoT. Esta adaptación es fundamental para dispositivos IoT, que a menudo requieren conectividad remota, una huella física mínima y la capacidad de operar en entornos donde la intervención humana es limitada o inexistente. La eSIM IoT está regida por especificaciones adicionales, como la SGP.31 de la GSMA, la cual aborda los requisitos particulares de los dispositivos conectados, incluyendo la provisión remota de eUICCs, la optimización de recursos y la garantía de seguridad en redes con limitaciones de interfaz y energía [23]. De esta manera, la eSIM IoT no sólo hereda las ventajas de la eSIM para el consumidor, sino que también añade características críticas para soportar la masificación de dispositivos IoT, facilitando la interoperabilidad, la gestión remota y el despliegue a gran escala en diversos sectores como la industria, la salud y la domótica.

El 3GPP también ha contribuido a la estandarización de la eSIM, enfocándose en aspectos relacionados con la seguridad y la gestión de

identidades en redes móviles, asegurando su interoperabilidad y funcionamiento en diferentes entornos de red.

Entre sus principales ventajas destacan la facilidad para cambiar de operador sin intervención física, el aprovechamiento del espacio interno del dispositivo y una mayor resistencia frente a factores ambientales, al eliminar la ranura externa

La adopción de la eSIM está transformando la forma en que los usuarios y los dispositivos se conectan a las redes móviles, ofreciendo una mayor comodidad y eficiencia en la gestión de la conectividad.

2.4. Estándares de la GSMA

La GSMA ha desarrollado una serie de estándares destinados a facilitar la implementación, administración y seguridad de la eSIM en dispositivos IoT, reconociendo las particularidades y restricciones propias de este entorno. En este contexto, el estándar SGP.31 se centra en la arquitectura y en los requisitos para la provisión remota de eUICCs en dispositivos IoT. Este estándar aborda aspectos críticos como la gestión de perfiles, la provisión remota y la actualización segura de la eSIM, considerando las limitaciones en conectividad, energía y capacidad de procesamiento propias de muchos dispositivos IoT [23]. La implementación de SGP.31 permite a los operadores de red y fabricantes de dispositivos garantizar una activación y administración remota eficiente y segura, esencial para la masificación de la conectividad en IoT.

Complementariamente, el estándar SGP.32 ofrece directrices adicionales que profundizan en los aspectos de seguridad, interoperabilidad y comunicación en la integración de eSIM para entornos IoT. Este estándar se orienta a establecer protocolos de intercambio de información y mecanismos de autenticación que aseguran que los perfiles de operador y la información crítica gestionada por la eSIM se mantengan protegidos ante posibles amenazas, facilitando la conformidad con normativas internacionales y promoviendo la integración de diferentes sistemas y plataformas [29].

Por otro lado, el conjunto de especificaciones conocido como IoT SAFE (IoT Subscriber Authenticity for Secure End-to-end) se ha diseñado para reforzar la seguridad en el ecosistema IoT mediante protocolos robustos de autenticación y autorización. IoT SAFE establece mecanismos de verificación de identidad y autenticidad en cada transacción y comunicación entre dispositivos, lo que resulta

crucial para prevenir accesos no autorizados y garantizar la integridad de los datos en un entorno cada vez más interconectado [30].

En conjunto, estos estándares de la GSMA –SGP.31, SGP.32 e IoT SAFE ofrecen un marco integral que no solo posibilita la gestión remota y la actualización segura de perfiles en dispositivos IoT, sino que también asegura la interoperabilidad y la protección de la información.

2.4.1 SGP.31

El estándar SGP.31 eSIM IoT Architecture and Requirements define la arquitectura y los requisitos necesarios para la provisión remota de eSIMs en dispositivos IoT que presentan restricciones de red y/o interfaz de usuario (UI). Su propósito principal es habilitar la interoperabilidad global en los escenarios de despliegue de IoT y establecer un marco seguro para la gestión remota de suscripciones.

Dado que los dispositivos IoT suelen operar en entornos con bajo ancho de banda, consumo energético reducido y capacidades de procesamiento limitadas, el estándar propone un modelo optimizado para la provisión, activación, desactivación y eliminación de perfiles de eSIM, asegurando que la conectividad se mantenga estable y segura durante todo el ciclo de vida del dispositivo.

2.4.1.1. Principios Fundamentales del SGP.31

El estándar SGP.31 define una serie de principios esenciales para asegurar una implementación de eSIMs en IoT que sea tanto segura como eficiente. En primer lugar, establece que toda gestión de perfiles como la descarga, activación o eliminación debe realizarse mediante conexiones autenticadas, preservando la integridad del proceso.

Además, contempla las limitaciones de muchos dispositivos IoT que operan en redes de baja capacidad, como LPWAN o NB-IoT, permitiendo la gestión de eSIMs sin necesidad de utilizar protocolos exigentes como TCP/IP. Otro aspecto clave es la eficiencia en las comunicaciones: se busca minimizar las operaciones de red y permitir que los dispositivos reciban instrucciones incluso después de largos periodos de inactividad. Para evitar la saturación de la red, se proponen mecanismos como el uso de notificaciones o consultas periódicas, que permiten

acceder a nuevas configuraciones sin generar tráfico innecesario. Finalmente, se refuerza la necesidad de garantizar la confidencialidad, autenticación e integridad de los datos en todas las transacciones, consolidando un entorno de gestión remoto robusto y confiable.

2.4.1.2. Arquitectura del SGP.31 para eSIM en IoT

El modelo de arquitectura definido en el estándar SGP.31 se basa en el esquema del SGP.21, pero introduce modificaciones específicas para entornos IoT. Los principales componentes son:

El estándar propone dos arquitecturas, según la disposición del IPA, dichas arquitecturas se describen a continuación.

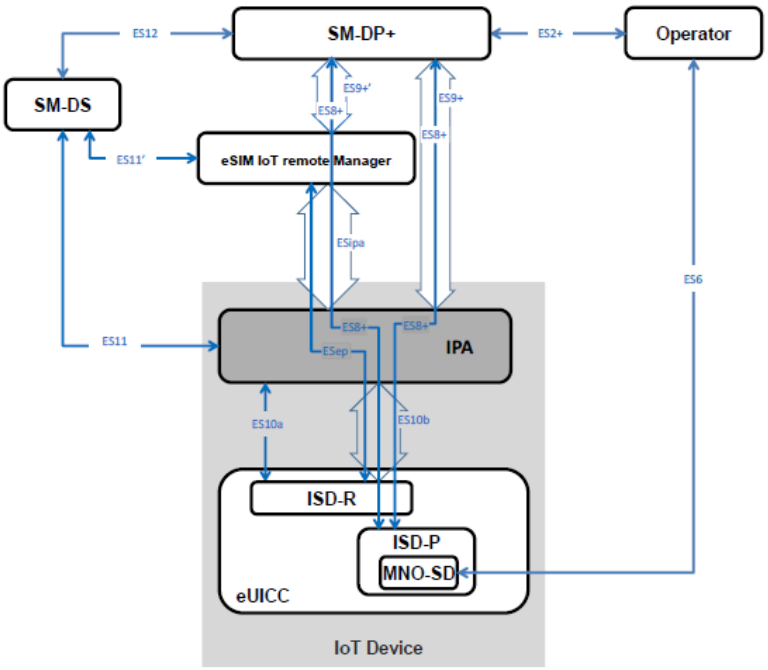

Figura 2.1. Arquitectura eSIM IoT, IPA en el dispositivo IoT [23]

La primera arquitectura, representada en la Figura 2.1, sitúa el IoT Profile Assistant (IPA) como un componente de software que reside directamente en el dispositivo IoT, pero fuera del chip seguro de la eUICC. En este modelo, el IPA es responsable de orquestar la gestión de perfiles, comunicándose tanto con los dominios de seguridad internos de la eUICC (como el ISD-R) como con entidades

de red externas. Este enfoque otorga flexibilidad al fabricante del dispositivo, pero delega parte de la lógica de seguridad al software del propio equipo.

Por otro lado, el estándar contempla una segunda configuración para reforzar la seguridad, como se muestra a continuación en la Figura 2.2. En esta arquitectura alternativa, el IPA se integra directamente dentro del perímetro seguro de la eUICC (IPAe).

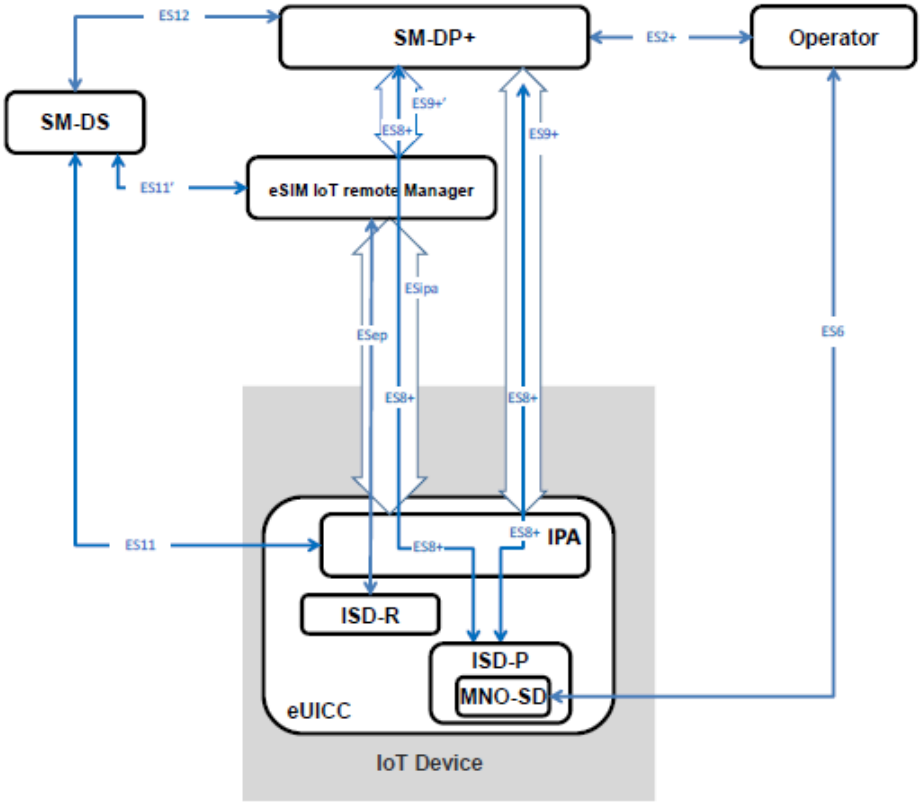

Figura 2.2. Arquitectura eSIM IoT, IPA en la eUICC [23]

- **eUICC (Embedded Universal Integrated Circuit Card):** Tarjeta integrada en el dispositivo IoT que contiene los perfiles de suscripción. Permite la provisión remota de perfiles mediante un canal seguro con los gestores de suscripciones.

- **SM-DP+ (Subscription Manager Data Preparation):** Responsable de la generación y entrega de perfiles de eSIM a los dispositivos IoT. Establece un canal seguro con la eUICC para transferir perfiles.

- **eIM (eSIM IoT Remote Manager):** Actúa como intermediario entre la infraestructura de red y los dispositivos IoT para gestionar perfiles de manera remota. Puede operar como entidad independiente o formar parte de plataformas de gestión de dispositivos.

- **IPA (IoT Profile Assistant):** Facilita la provisión de perfiles en dispositivos IoT mediante la comunicación con el SM-DP+. Puede residir dentro del dispositivo IoT (IPAd) o en la eUICC (IPAe).

- **SM-DS (Subscription Manager Discovery Server)**: Proporciona un mecanismo para que las eSIMs descubran servidores de suscripción disponibles sin intervención manual.

En la figura 2.3 se describe la arquitectura interna de una eUICC con IoT Profile Assistant (IPA) embebido.

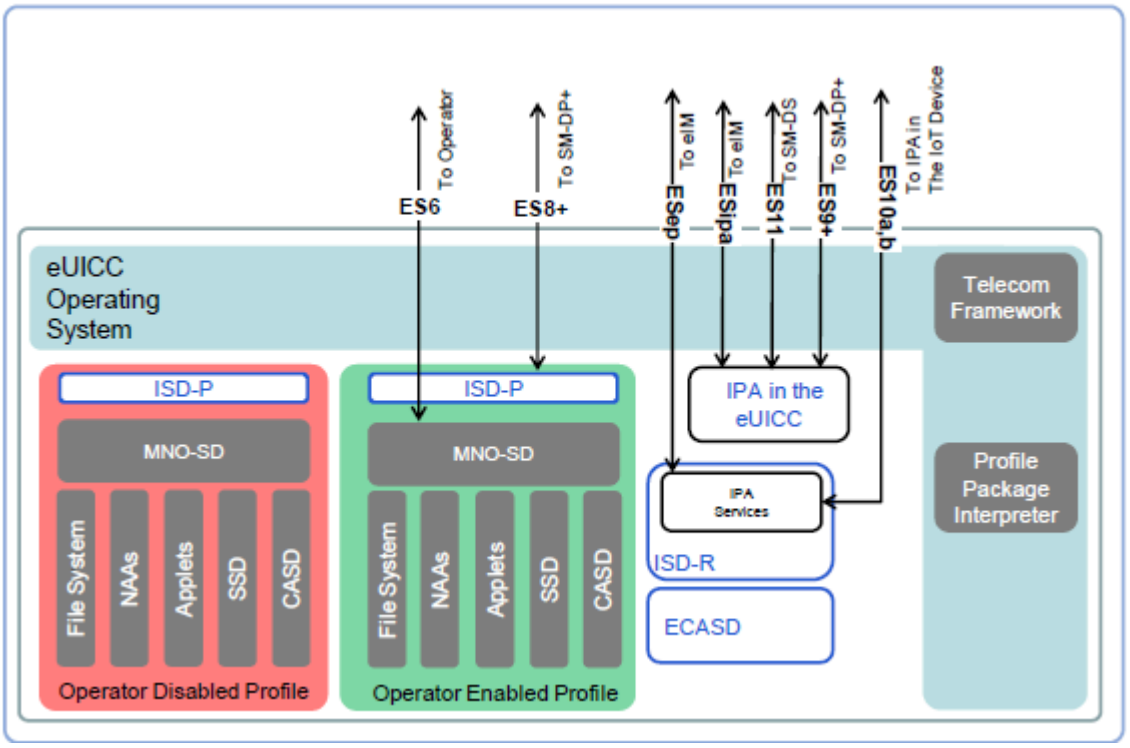

Figura 2.3. Arquitectura de eUICC con IPA embebido [23]

Con el objetivo de asegurar tanto la protección como la interoperabilidad de las eSIMs en entornos IoT, el estándar SGP.31 incorpora una serie de requisitos de seguridad fundamentales. Entre ellos, se establece que todas las operaciones

realizadas sobre la eUICC deben estar firmadas digitalmente y ser verificadas por entidades autorizadas, garantizando así la autenticación y autorización adecuadas. Para evitar ataques de repetición, el estándar incluye mecanismos específicos que previenen la reutilización fraudulenta de mensajes en las comunicaciones. Asimismo, se exige el uso de criptografía avanzada basada en cifrado asimétrico, con el fin de preservar la integridad y autenticidad de los datos transmitidos. Finalmente, se incorporan mecanismos de resiliencia en la gestión de perfiles, como el perfil de respaldo (Fallback Profile) o el mecanismo de reversión (Rollback Mechanism), que permiten restaurar configuraciones anteriores en caso de fallos durante el aprovisionamiento o pérdida de conectividad.

2.4.2. SGP.32

El estándar SGP.32 (eSIM IoT Technical Specification) define los detalles técnicos de la arquitectura eSIM IoT, proporcionando una base operativa para la gestión remota de perfiles eSIM en dispositivos IoT con restricciones de red y sin interfaz de usuario (NCDs/UICDs). Esta especificación describe en profundidad los protocolos de comunicación, estructura de datos, seguridad, interfaces y mecanismos de autenticación utilizados para el aprovisionamiento y gestión de perfiles en entornos IoT.

2.4.2.1. Arquitectura Técnica y Flujo de Datos en SGP.32

El estándar SGP.32 define un flujo de comunicación basado en varias interfaces clave entre los componentes del ecosistema eSIM IoT. Los datos se transfieren en formato ASN.1 y JSON, codificados en TLV (Tag-Length-Value) y protegidos con cifrado de clave pública.

Intercambio de Datos en la Gestión de Perfiles:

1. **Solicitud de Descarga de Perfil (ES9+ y ES9+')**

- El IPA (IoT Profile Assistant) o el eIM (eSIM IoT Remote Manager) inicia una solicitud de descarga de perfil al SM-DP+ (Subscription Manager Data Preparation+).
- Se establece un canal TLS/DTLS entre el IPA/eIM y el SM-DP+.
- Se verifica la autenticidad del dispositivo usando certificados X.509 y claves ECDSA.

2. **Descarga del Perfil (ES8+)**

- El SM-DP+ transfiere el paquete de perfil encriptado a la eUICC mediante la interfaz ES8+.
- La eUICC verifica la firma digital del paquete y procede con la instalación.

3. **Habilitación y Gestión del Perfil (ES10a y ES10b)**

- El IPA envía un comando para activar o desactivar un perfil en la eUICC.
- Se usa un mecanismo de transacciones atómicas para evitar estados inconsistentes.

4. **Eliminación o Modificación del Perfil (ESep, ES12)**

- Los perfiles pueden ser deshabilitados, eliminados o modificados a través del eIM.
- Se asegura la integridad mediante firmas digitales y autenticación mutua.

2.4.2.2. Protocolos de Comunicación en SGP.32

SGP.32 admite múltiples protocolos optimizados para IoT, seleccionados según las capacidades del dispositivo y la red disponible:

1. **HTTP sobre TLS (HTTPS) - Protocolo Primario**

   - Utilizado para la comunicación entre SM-DP+, eIM e IPA.
   - Usa TLS 1.2/1.3 con Perfect Forward Secrecy (PFS) para proteger los datos.
   - Se implementa autenticación mutua mediante certificados X.509.

2. **CoAP (Constrained Application Protocol) sobre DTLS**

   - Protocolos ligeros diseñados para LPWAN, NB-IoT y redes restringidas.
   - Admite el uso de DTLS 1.2/1.3 con cifrado AES-GCM para seguridad.

- Usa mensajes conformables (CON) y no confirmables (NON) para minimizar el consumo de ancho de banda.

3. **MQTT (Message Queuing Telemetry Transport) sobre TLS**

- Se usa en entornos donde el eIM actúa como broker MQTT.
- Soporta QoS 0, 1 y 2, garantizando entrega de mensajes en dispositivos con conectividad intermitente.
- Implementa autenticación mediante certificados X.509 y tokens JWT.

2.4.2.3. Gestión de Seguridad en SGP.32

La especificación SGP.32 incorpora mecanismos avanzados de seguridad para garantizar la integridad, autenticidad y confidencialidad de las transacciones.

1. **Mecanismos de Autenticación y Cifrado**

- Se utilizan certificados X.509 para autenticar todas las entidades.
- Se admite cifrado de curva elíptica (ECC) con ECDSA y ECDH.
- Todas las claves privadas deben almacenarse en un HSM (Hardware Security Module) o TPM (Trusted Platform Module).

2. **Protección contra Ataques de Repetición**

Se usa un contador de transacciones para evitar la repetición de mensajes antiguos. Se implementa un Nonce único en cada sesión de comunicación.

3. **Gestión de Certificados y Firmas Digitales**

Certificado	Uso
CERT.EIM.ECDSA	Certificado de firma digital del eIM
CERT.EIM.TLS	Certificado para autenticación TLS/DTLS
CERT.EUICC.ECDSA	Certificado de autenticación de la eUICC

Tabla 2.2. Certificados SGP32.

Todos los certificados mostrados en la tabla 2.2 deben seguir el formato X.509 v3 y pueden ser revocados mediante OCSP o CRLs.

2.4.2.4. Gestión de Estados de Perfiles y Mecanismos de Recuperación

SGP.32 introduce un conjunto de mecanismos para la gestión dinámica de estados de perfiles en la eUICC:

1. Estados de los Perfiles, ver tabla 2.3

Estado	Descripción
Provisioning Profile	Perfil utilizado para la activación inicial
Operational Profile	Perfil activo que permite conectividad
Fallback Profile	Perfil de respaldo en caso de falla
Test Profile	Perfil para pruebas y certificación

Tabla 2.3. Gestión de perfiles

2. Mecanismos de Recuperación

- Rollback Mechanism: Permite restaurar un perfil anterior en caso de falla en la provisión.
- Fallback Mechanism: Activa automáticamente un perfil de respaldo si el perfil principal falla.
- Immediate Profile Enablement: Permite habilitar un perfil inmediatamente después de su descarga.

El estándar SGP.32 proporciona un marco técnico detallado para la provisión remota y gestión segura de eSIMs en dispositivos IoT. Incorpora protocolos optimizados para redes IoT, mecanismos avanzados de seguridad y una arquitectura flexible para soportar diferentes entornos de conectividad.

2.5. Estado del arte y trabajos relacionados

En los últimos años, la convergencia de las tecnologías blockchain, IoT y las redes de nueva generación ha generado un campo de investigación muy activo. Diversos trabajos exploran cómo la descentralización puede resolver desafíos históricos de seguridad, autenticación e interoperabilidad en sistemas de telecomunicaciones. Esta sección analiza los trabajos relacionados más relevantes, agrupándolos en tres áreas temáticas principales: (1) la autenticación descentralizada y gestión de la identidad, (2) la securización del ciclo de vida y

aprovisionamiento de perfiles eSIM, y (3) el uso de blockchain como herramienta para la trazabilidad y auditoría. Este mapa del estado del arte permitirá situar la contribución específica de este trabajo.

2.5.1. Autenticación Descentralizada y Gestión de Identidad

Una de las áreas con mayor exploración es el uso de blockchain para superar las limitaciones de los modelos de autenticación centralizados, que presentan puntos únicos de fallo y dependen de un tercero de confianza. En este ámbito, varias investigaciones proponen esquemas de autenticación descentralizada.

Un ejemplo destacado es DecAuth, un esquema propuesto por Mohanta et al. para la autenticación de dispositivos IoT aprovechando la plataforma Ethereum [32]. El objetivo principal de este trabajo es proponer una alternativa a los sistemas centralizados, que son vulnerables a ataques de denegación de servicio (DoS) [76] y donde el usuario debe confiar plenamente en el servidor. La solución se materializa en un contrato inteligente que almacena identificadores de usuario (user IDs) y sus direcciones de billetera asociadas. Una de sus características más notables es el uso de un par de claves con roles diferenciados: una clave de autenticación (authKey), que es la que se utiliza en el día a día para firmar mensajes y autenticarse en los servicios, y una clave de recuperación (recoveryKey), que actúa como una "llave maestra" para gestionar la cuenta, permitiendo al usuario restaurar el acceso si la authKey se ve comprometida [32].

El flujo de autenticación es análogo a un "Log In with Facebook" descentralizado: un servicio genera un mensaje único que el usuario firma con su authKey a través de una herramienta como MetaMask [77]; el servicio luego verifica esta firma para validar la sesión. Los autores implementaron un prototipo utilizando una red de prueba de Ethereum (Ganache), el framework Truffle [78] para la gestión del contrato escrito en Solidity y la librería web3.js para la integración con un frontend en ReactJs. El análisis de seguridad concluye que esta arquitectura es resistente a ataques comunes como Man-in-the-Middle (MITM), ataques de repetición e impersonación, gracias al uso de firmas digitales y hashing criptográfico. Además, señalan que, si bien la creación de cuentas en Ethereum puede tener latencia, la verificación de usuarios es rápida y el sistema es altamente escalable [32].

Esta misma filosofía se ha aplicado a entornos de telecomunicaciones más complejos como las redes 5G. El protocolo estándar 5G-AKA [60], definido en la especificación TS 33.501, presenta vulnerabilidades críticas derivadas de su arquitectura centralizada, que depende de las entidades Authentication Server Function (AUSF) y Unified Data Management (UDM) [37]. Entre los fallos identificados se encuentran la exposición a ataques de denegación de servicio (DoS/DDoS), ataques de linkability que permiten el rastreo de dispositivos a través de mensajes de error (MAC_FAIL y SYNC_FAIL), y la falta de secreto perfecto hacia adelante [36]. Para solucionar estos problemas, la investigación de Chow y Ma introduce el esquema 5GSBA (Secure Blockchain-based Authentication and Key Agreement) [36]. La propuesta descentraliza radicalmente la función de autenticación, trasladándose desde los servidores centrales directamente a las estaciones base (gNBs). Para ello, se utiliza una blockchain privada como un repositorio de datos de suscriptores distribuido entre todos los gNBs, eliminando el punto único de fallo.

La arquitectura de 5GSBA se fundamenta en varios mecanismos de seguridad robustos. Primero, para evitar la suplantación, cada dispositivo (USIM) se provisiona con una clave secreta de un solo uso (one-time hash secret). En la blockchain solo se almacena el hash de dicha clave, de modo que cuando un dispositivo presenta su clave secreta para autenticarse, el gNB puede verificarla calculando el hash y comparándolo con el registro inmutable, sin que la clave secreta viaje por la red o sea expuesta en el ledger [36]. Segundo, para proteger la privacidad, el protocolo hace obligatorio el uso de SUCI (Subscription Concealed Identifier), que cifra la identidad permanente del dispositivo (SUPI). Tercero, para evitar los ataques de linkability, se elimina el uso de números de secuencia, propensos a fallos de sincronización, y se sustituyen por un intercambio de claves basado en Criptografía de Curva Elíptica (ECDH) [79], garantizando así el secreto perfecto hacia adelante.

La validez y seguridad del protocolo fueron comprobadas rigurosamente mediante dos métodos: un análisis de lógica formal con lógica BAN (Burrows-Abadi-Needham) [80] y una verificación automatizada con la herramienta Scyther [81]. La evaluación de rendimiento demostró que, aunque 5GSBA introduce una pequeña sobrecarga computacional (aproximadamente 1.67 ms adicionales), esta es perfectamente asumible y se justifica por las significativas mejoras en seguridad que ofrece, especialmente su alta resiliencia frente a ataques de inundación DDoS [36].

2.5.2. Securización del Ciclo de Vida y Aprovisionamiento de Perfiles eSIM

Con el crecimiento masivo de dispositivos IoT, los métodos tradicionales de configuración manual se han vuelto inadecuados. La transición hacia la eSIM ha introducido el Aprovisionamiento Remoto de SIM (RSP), que, aunque flexible, presenta nuevos desafíos de seguridad.

Para abordar este problema, existen trabajos que proponen arquitecturas integrales para un aprovisionamiento seguro y sin intervención (zero-touch), como el descrito por Krishnan et al. en su trabajo "eSIM and blockchain integrated secure zero-touch provisioning for autonomous cellular-IoTs in 5G networks" [31]. Su propuesta ataca la ineficiencia y la inseguridad de la provisión manual a gran escala.

Para ello, diseñan un framework integrado, denominado SIeSIM, que combina cuatro tecnologías clave: 1) eSIM junto con el protocolo IoT-SAFE de la GSMA [30], utilizando la SIM como "Raíz de Confianza" (Root of Trust) para la gestión segura de la identidad; 2) Blockchain, utilizando contratos inteligentes en Ethereum como un registro descentralizado e inmutable para almacenar y verificar los manifiestos de red; 3) Redes Definidas por Software (SDN), para la orquestación dinámica y programable de los recursos de red y las políticas de seguridad; y 4) Protocolos de comunicación segura como TLS/DTLS. La arquitectura fue validada mediante simulaciones en el simulador COOJA [82] y un montaje experimental con una puerta de enlace IoT conectada a Microsoft Azure.

La evaluación de rendimiento arrojó resultados muy positivos, destacando una reducción drástica del Tiempo de Aprovisionamiento (Time-To-Provision) a aproximadamente 240 milisegundos, lo que supera en un 320% a los métodos manuales. Este trabajo demuestra el potencial de combinar múltiples tecnologías emergentes para crear una solución completa y eficiente [31].

Un trabajo particularmente relevante que se centra en una vulnerabilidad específica del ciclo de vida es el que propone el SIM Profile Transparency Protocol (SPTP), de Ahmed et al. [38]. Esta investigación aborda un riesgo crítico del ecosistema RSP: la posibilidad de que un servidor de aprovisionamiento (SM-DP+)

comprometido pueda clonar perfiles de SIM genuinos (duplicando el IMSI y las claves secretas) y promocionarlos de forma fraudulenta. Para mitigarlo, SPTP propone un mecanismo de auditoría basado en la transparencia total de las operaciones de provisión. La solución introduce dos nuevos actores:

Private Index Calculator (PIC): Un componente que genera un índice privado para cada IMSI utilizando una Función Aleatoria Verificable (VRF). La VRF crea un índice que no revela el IMSI original (protegiendo la privacidad), pero que puede ser verificado públicamente por partes autorizadas.

Transparency Ledger (T): Un registro inmutable de solo adición, implementado sobre una blockchain permisionada, que almacena una "vinculación" (binding) por cada operación de provisión. Este registro asocia el índice del PIC con un hash del International Mobile Subscriber Identity (IMSI) y el identificador de la eUICC (EID), además de otros metadatos como el identificador del servidor.

El protocolo exige que el servidor registre esta vinculación en el ledger antes de que una eUICC acepte el perfil, y esta a su vez verifica dicho registro. Esto permite a los operadores (MNOs) monitorizar el ledger y detectar cualquier provisión no autorizada de perfiles que usen su rango de IMSIs. La propuesta fue validada con un análisis formal usando la herramienta ProVerif y mediante un prototipo implementado en Solidity sobre una blockchain Quorum (un fork de Ethereum enfocado en la privacidad) [83]. Los resultados demostraron que el impacto en el rendimiento es mínimo (un aumento de solo 107 bytes en el tamaño del perfil y una latencia de registro inferior a 3 segundos), validando su aplicabilidad práctica [38].

2.5.3. Blockchain como Herramienta para la Trazabilidad y Auditoría

Un beneficio transversal que la literatura atribuye a blockchain es su capacidad para crear un registro de eventos auditable, transparente e inmutable. Tanto el protocolo SPTP [38], con su Transparency Ledger diseñado específicamente para la auditoría, como los trabajos sobre aprovisionamiento en 5G [31], utilizan la blockchain para registrar y verificar transacciones, garantizando la integridad de todo el ciclo de vida del dispositivo.

De forma complementaria, la guía de implementación IoT SAFE de la GSMA consolida la visión de la industria de posicionar la SIM como un "Elemento Raíz de

Confianza" (Root of Trust) [30]. El documento, titulado "Common Implementation Guide to Using the SIM as a 'Root of Trust' to Secure IoT Applications", establece una guía práctica para aprovechar la tarjeta SIM (o eSIM) como un componente de hardware seguro. El principal problema que aborda es que muchos dispositivos IoT, especialmente los de bajo coste, carecen de mecanismos de protección nativos para sus credenciales, lo que los hace vulnerables. La guía propone utilizar la SIM, por su naturaleza segura y estandarizada, para guardar credenciales críticas y ejecutar funciones criptográficas.

De manera crucial para este TFM, el anexo informativo de la guía IoT SAFE sugiere explícitamente una arquitectura alternativa para servicios IoT seguros basada en tecnología blockchain. Esta arquitectura, descrita en el Anexo A del documento, utiliza la blockchain como un repositorio confiable para almacenar y recuperar claves públicas, pudiendo complementar o incluso sustituir a una Infraestructura de Clave Pública (PKI) tradicional.

En este modelo, la blockchain actúa como un punto de confianza compartido entre diferentes dominios (por ejemplo, el operador móvil y un proveedor de servicios IoT), permitiendo la publicación verificada de claves públicas y habilitando casos de uso como la notarización de datos recogidos por sensores remotos. El sistema propuesto interactuaría con plataformas OTA para la gestión remota de los applets y claves en la SIM, mientras la blockchain proporciona la capa de confianza para la infraestructura de claves. Esta propuesta de la GSMA valida el enfoque de utilizar la SIM como ancla de seguridad en un ecosistema descentralizado y aporta un fuerte respaldo industrial a la línea de investigación de este trabajo [30].

2.5.4. Posicionamiento de este Trabajo

Tras analizar el estado del arte y los trabajos relacionados, este trabajo se posiciona como una propuesta que sintetiza y extiende varias de las ideas investigadas.

A diferencia de trabajos como DecAuth [32] o 5GSBA [36] que se centran principalmente en el protocolo de autenticación, este proyecto aborda el ciclo de vida completo de la gestión del perfil eSIM, incluyendo no sólo la identidad inicial, sino también operaciones posteriores como el cambio de operador.

Inspirado en el enfoque de SPTP [38], este trabajo también utiliza una blockchain para la trazabilidad y la gestión operativa de perfiles de red. Sin embargo, amplía su alcance más allá de la prevención de la clonación de perfiles para incluir un abanico más amplio de eventos operativos, creando así un marco de gestión y auditoría más completo.

Finalmente, tomando como referencia la visión de la GSMA en IoT SAFE [30], las arquitecturas propuestas en este trabajo exploran de manera práctica cómo la eUICC, con un IPA embebido, puede actuar como ese "Elemento Raíz de Confianza" que interactúa de forma segura y directa con una infraestructura descentralizada.

En resumen, estos trabajos aportan un modelo de gestión integral que no solo securiza, sino que también aporta interoperabilidad y transparencia al ecosistema eSIM/IoT, conectando las necesidades de la industria con las capacidades de la tecnología blockchain.

Capítulo 3 - Metodologías y Tecnologías

Para llevar a cabo este trabajo se aplicaron un conjunto de metodologías que combinaron investigación documental, desarrollo de prototipos, validación técnica, e investigación social a profesionales del sector. Esta combinación permitió abordar el problema desde distintos ángulos, garantizando tanto la investigación teórica como la aplicabilidad práctica de la solución propuesta.

3.1. Revisión documental y análisis de estándares

El proyecto comenzó con una fase de investigación basada en la revisión de documentación técnica y bibliografía especializada. Se estudiaron en detalle los estándares definidos por la GSMA, especialmente los documentos SGP.31 y SGP.32. Esta revisión fue clave para entender el funcionamiento de los procesos de gestión de eSIMs en dispositivos IoT y para identificar las oportunidades de mejora en términos de trazabilidad, seguridad e interoperabilidad. Además, se consultaron artículos académicos, white papers y casos de uso reales que contribuyeron a contextualizar la propuesta dentro del ecosistema actual de IoT.

3.2. Desarrollo del prototipo

Una vez definidos los objetivos técnicos, se procedió a la construcción de una prueba de concepto (PoC) para validar la viabilidad de los componentes más innovadores de la arquitectura. El objetivo de este prototipo no era replicar el ecosistema completo de telecomunicaciones, sino demostrar la funcionalidad del núcleo de la propuesta: la interacción entre un backend orquestador, una base de datos de estado y una red blockchain para registrar eventos de forma segura e inmutable.

Para ello, se adoptó un enfoque de desarrollo iterativo e incremental. El trabajo se organizó en fases progresivas, lo que permitió construir y validar cada componente de la solución de manera controlada antes de integrar en flujos de trabajo más complejos. El ciclo de desarrollo para cada una de las funcionalidades clave (como el registro de identidad o el cambio de operador) siguió los siguientes pasos:

1. Diseño de la lógica del contrato inteligente: Se definieron las funciones, eventos y estructuras de datos necesarias en el smart contract para gobernar cada operación en la blockchain.
2. Implementación de la lógica de negocio en el Backend: Se desarrolló el endpoint de la API correspondiente en el backend, programando la lógica para validar las peticiones, interactuar con la base de datos de estado y construir y enviar la transacción al contrato inteligente.
3. Simulación de los actores del ecosistema: Se crearon scripts para simular las peticiones que realizan los componentes externos, como el IoT Profile Assistant (IPA), permitiendo probar los flujos de comunicación.
4. Pruebas Unitarias y Funcionales: Se aplicaron pruebas a dos niveles: pruebas unitarias para cada función individual del backend y del contrato, y pruebas funcionales de extremo a extremo para verificar la correcta ejecución de los flujos.

Este enfoque metodológico aseguró que la prueba de concepto validará de manera efectiva la hipótesis central del trabajo, demostrando que es técnicamente factible orquestar y registrar el ciclo de vida de una eSIM utilizando una capa de confianza a través de blockchain.

3.3 Aplicación de encuesta técnica

Además del desarrollo técnico, se incorporó una validación cualitativa mediante la aplicación de una encuesta dirigida a personas con conocimientos técnicos en IoT, telecomunicaciones y ciberseguridad. Esta encuesta tenía como objetivo recopilar opiniones sobre la viabilidad, utilidad y escalabilidad de la solución propuesta. Para seleccionar a los participantes se utilizó el método de muestreo por bola de nieve (snowball sampling), lo que permitió ampliar la muestra inicial a través de contactos recomendados por los propios encuestados. De esta manera, se logró incluir perfiles variados y representativos del sector, evitando que la muestra quedará limitada a conocidos directos. Los registros obtenidos de la encuesta realizada se compartieron a través de Zenodo [54] que es un plataforma de acceso público que permite a los investigadores compartir datos e informes, la misma fue desarrollada por el programa europeo OpenAire y es mantenida por CERN.

3.4. Tutorías y supervisión académica

Durante todo el proceso se llevaron a cabo sesiones periódicas de tutoría con el director de este trabajo. Estas sesiones sirvieron como espacios de revisión, orientación metodológica y validación técnica. Gracias a este acompañamiento continuo fue posible mantener una coherencia clara entre los objetivos iniciales, el desarrollo del prototipo y los resultados obtenidos. Además, las tutorías permitieron reajustar el enfoque del proyecto en función de la evolución del contexto o la aparición de nuevas oportunidades técnicas.

3.5. Herramientas y tecnologías utilizadas

Para el desarrollo del prototipo y la elaboración de este documento se utilizó un conjunto de herramientas de software, plataformas y lenguajes de programación. A continuación, se detallan los componentes tecnológicos empleados, agrupados por su función dentro del proyecto.

Desarrollo del Backend y Base de Datos:

- **Python:** Se utilizó como lenguaje de programación principal para el desarrollo del backend [61]. Se eligió por su sintaxis clara, su amplio ecosistema de librerías y su fuerte soporte para el desarrollo web y la interacción con blockchains.
- **Flask:** Es un microframework para Python [62] que se empleó para construir la API REST del backend. Permitió desarrollar rápidamente los endpoints necesarios para la gestión de eventos y la comunicación con los demás componentes de la arquitectura.
- **SQLite3:** Para la persistencia de datos en el prototipo, se utilizó SQLite3 [63], un motor de base de datos relacional sin servidor. Su simplicidad fue ideal para gestionar el estado de los dispositivos y el historial de operaciones en un entorno de desarrollo local.
- **WSL (Windows Subsystem for Linux):** Se empleó como entorno de desarrollo para asegurar la compatibilidad y facilitar la integración de herramientas de línea de comandos de Linux, como las utilizadas para la gestión de la blockchain, en un sistema operativo Windows [64].

Desarrollo Blockchain (Contratos Inteligentes e Interacción):

- **Solidity:** Es el lenguaje de programación orientado a objetos utilizado para escribir los contratos inteligentes [49]. Está diseñado para ejecutarse en la

Máquina Virtual de Ethereum (EVM) y es el estándar de facto para el desarrollo en esta plataforma.

- **Hardhat:** Se utilizó como entorno de desarrollo principal para Ethereum [65]. Permitió compilar, desplegar, probar y depurar los contratos inteligentes en una red local simulada, agilizando significativamente el ciclo de desarrollo.
- **Remix IDE:** Se usó como herramienta complementaria para la escritura y validación rápida de contratos inteligentes [66]. Al ser una plataforma web, facilitó la compilación y el despliegue de versiones preliminares del código.
- **Web3.py:** Es la librería de Python que se utilizó en el backend para interactuar con los nodos de Ethereum [67]. Permitió enviar transacciones a los contratos inteligentes (por ejemplo, para registrar un evento) y leer datos de la blockchain.

Herramientas de Desarrollo, Modelado y Colaboración:

- **Visual Studio Code (VS Code):** Fue el editor de código principal utilizado para el desarrollo tanto del backend en Python como los diagramas de secuencia.
- **GitHub:** Se empleó como plataforma para alojar el código fuente del proyecto. El repositorio, disponible públicamente, se distribuye bajo licencia MIT "La licencia MIT, es una licencia de software libre permisiva, que otorga a los usuarios la libertad de usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar y vender copias del software, siempre y cuando se mantenga el aviso de derechos de autor y el texto de la licencia MIT". Para acceder al código fuente del proyecto, debe dirigirse al siguiente repositorio público.

  https://github.com/johmolin/TFM_MIOT_JM

- **UML (Unified Modeling Language):** Se utilizó para el modelado de los flujos de interacción del sistema [68].
- **Asistentes de IA:** Se contó con el apoyo de herramientas de inteligencia artificial como ChatGPT [69], Gemini [70] para agilizar el desarrollo y estructura del código del proyecto.
- **Google Docs:** Se utilizó como plataforma colaborativa para la redacción de la memoria del TFM y el intercambio de comentarios con el tutor [71].

Este capítulo ha detallado el enfoque metodológico integral que ha guiado el presente trabajo. La combinación de una exhaustiva revisión documental y de estándares sentó las bases teóricas, mientras que el desarrollo de un prototipo mediante un proceso iterativo permitió demostrar la viabilidad técnica de la

arquitectura propuesta. Finalmente, la validación cualitativa a través de una encuesta a expertos aportó una perspectiva crucial sobre la relevancia y aplicabilidad de la solución en el contexto industrial. El conjunto de tecnologías y herramientas seleccionadas fue fundamental para materializar cada una de estas fases, garantizando así que los resultados y conclusiones de este trabajo se sustenten tanto en lo académico como en la evidencia práctica.

Capítulo 4 - Arquitecturas Técnicas Propuestas.

En este capítulo se presentan dos arquitecturas técnicas alternativas para la integración del IoT Profile Assistant (IPA) en un sistema de gestión y provisión de perfiles eSIM en dispositivos IoT, conforme al estándar SGP.31 y 32 de la GSMA. Ambas soluciones abordan el desafío de gestionar remotamente los perfiles móviles (eSIM), mediante el uso de un backend que funcionará como orquestador entre todos los eventos generados desde el dispositivo y la red móvil contra la blockchain. La diferencia entre ambas propuestas radica en la ubicación funcional del módulo IPA, que puede residir en el propio dispositivo IoT o en la eUICC.

4.1. Blockchain e IPA (IoT Profile Assistant) en el dispositivo IoT

En esta propuesta, se adopta el modelo en el cual el IoT Profile Assistant (IPA) está embebido directamente en el dispositivo IoT. Este enfoque permite una mayor autonomía del dispositivo en la gestión de perfiles eSIM, optimizando procesos de operación como el aprovisionamiento remoto, la activación y la conmutación de perfiles, sin intervención directa del SM-SR (Subscription Manager Secure Routing).

El sistema integra un backend que actúa como orquestador lógico entre el dispositivo IoT, la red blockchain y la plataforma de provisión eSIM. Este backend ha sido prototipado, incluyendo lógica de operación desarrollada en smart contracts desplegados sobre una red blockchain. Así como lógica de orquestación mediante la creación de endpoints a través de API que permite la interacción directa entre todos los componentes de la arquitectura. Como puede ser registro de un dispositivo mediante la descarga de un perfil, cambios de operador, autenticación y verificación de identidad, entre otros.

A continuación, se muestra en la figura 4.1 la arquitectura propuesta.

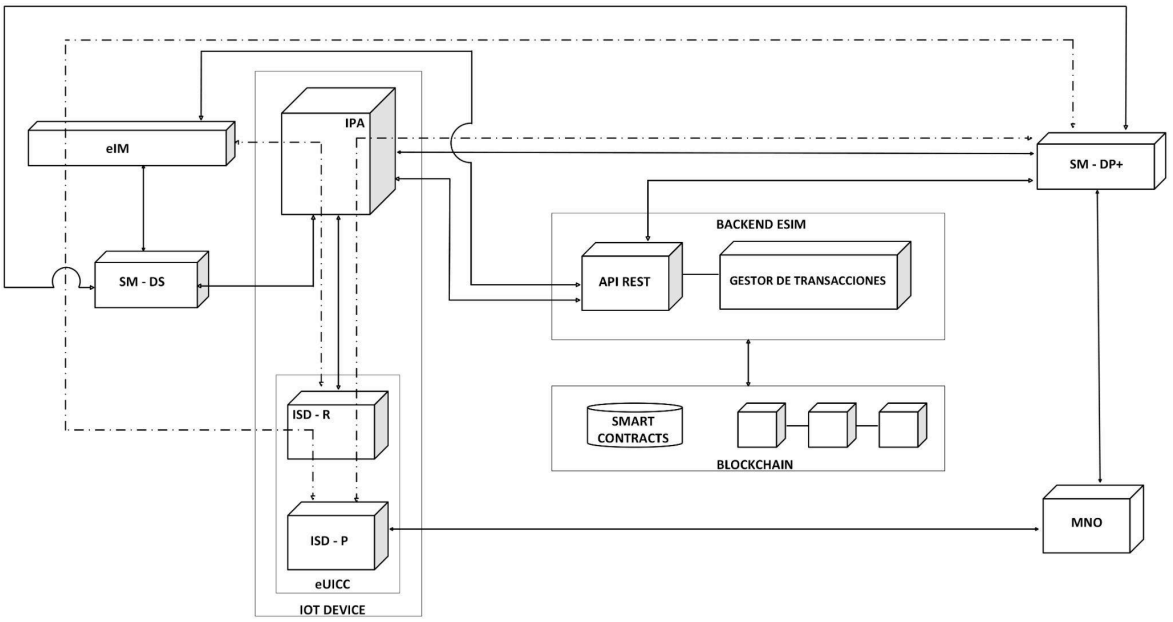

Figura 4.1. Blockchain e IPA (IoT Profile Assistant) en el dispositivo IoT

A continuación, se describe detalladamente cada uno de los componentes que forman parte de la arquitectura propuesta.

4.1.1. Componentes de la arquitectura.

4.1.1.1. IoT device

En la siguiente figura se muestra el IoT device componente que analizaremos a continuación:

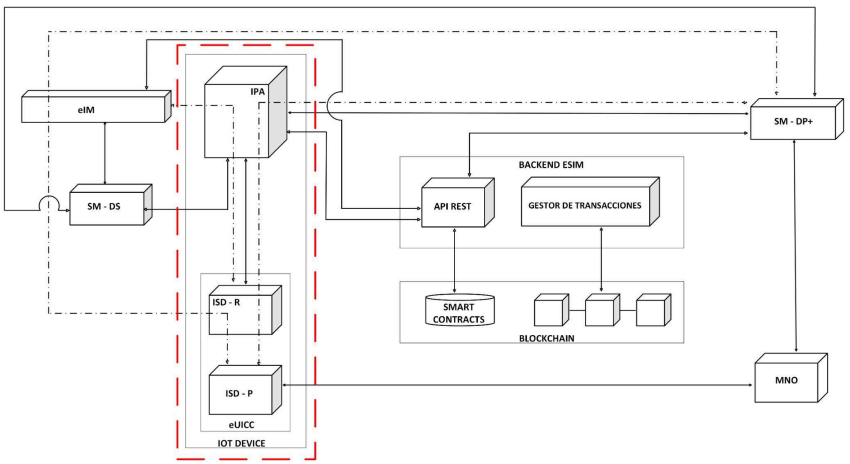

Figura 4.2. Bloque IoT Device

1. **Estructura interna del bloque.**

Este bloque (ver figura 4.2) representa un dispositivo IoT que integra un módulo eUICC (embedded Universal Integrated Circuit Card), que a su vez contiene:

- **ISD-R (Issuer Security Domain - Root):** Elemento de seguridad raíz que controla la gestión de perfiles y acceso a la tarjeta eUICC.
- **ISD-P (Issuer Security Domain - Profile):** Dominio que contiene el perfil del operador móvil que será activado.
- **IPA (IoT Profile Assistant):** Software embebido en el dispositivo que actúa como agente inteligente de control para la gestión de perfiles eSIM directamente desde este.

2. **Funciones del IPA**

El IPA ejecuta las siguientes funciones críticas:

- Selección y activación de perfiles almacenados en el eUICC.
- Gestión de eventos locales (cambio de operador, registro en la red, etc.).
- Comunicación bidireccional con el backend eSIM mediante API REST para registrar transacciones.
- Notificación de eventos al eIM (eSIM Intermediary Manager) para facilitar interacción con la SM-DP+.

3. **Protocolos**

- **HTTPS sobre TLS 1.3**: Se utiliza como canal seguro para las comunicaciones RESTful entre el IPA y el backend eSIM, garantizando confidencialidad, integridad y autenticación mutua. TLS 1.3 es la versión más actual del protocolo de seguridad en la capa de transporte, recomendado por la IETF (RFC 8446) debido a su menor latencia y mayor robustez criptográfica frente a versiones anteriores [39].

- **GSMA SGP.31 / SGP.32**: Estos estándares definen el marco funcional y las interfaces de comunicación para la gestión remota de perfiles eSIM en dispositivos IoT con capacidades LPWA. Específicamente, SGP.31 describe los requisitos técnicos y de arquitectura, mientras que SGP.32 detalla las APIs

y protocolos de control entre el IPA, el eIM, y el entorno seguro de la eUICC.

- **APDU / ISO/IEC 7816-4**: La comunicación de bajo nivel entre el IPA y las aplicaciones dentro de la eUICC (como ISD-R e ISD-P) se realiza mediante comandos APDU (Application Protocol Data Unit), estandarizados por la norma ISO/IEC 7816-4. Este estándar especifica la estructura y el comportamiento de los comandos de intercambio con tarjetas inteligentes, incluyendo la codificación, canales lógicos y mecanismos de respuesta [40].

- **Criptografía ECC / RSA (2048 o 4096 bits)**: Se emplean algoritmos de criptografía de clave pública para la autenticación mutua y la verificación de integridad en las comunicaciones seguras. RSA es ampliamente utilizado por su madurez y soporte generalizado, mientras que ECC (Elliptic Curve Cryptography) ofrece un nivel de seguridad equivalente con claves más pequeñas y mayor eficiencia computacional, según NIST [41][59].

- **JWT (JSON Web Tokens)**: Para las solicitudes entre el IPA y el backend eSIM, se utiliza el esquema de autenticación basado en JWT, que permite firmar y verificar tokens portables de acceso. JWT es un estándar abierto (RFC 7519) que facilita la autorización segura en sistemas distribuidos, manteniendo la integridad del contenido mediante firmas digitales con algoritmos como RS256 o ES256 [42].

4. Mecanismos de Seguridad

- Zona de ejecución segura (Trusted Execution Environment, TEE) donde reside el IPA.
- Autenticación basada en certificados X.509 emitidos por autoridad raíz del operador IoT.
- Validación de firma digital para cualquier perfil antes de ser activado.
- Protección contra ataques de repetición mediante nonces y timestamps en las peticiones al backend.

5. Flujo de Comunicación

1. El IPA se activa en el dispositivo y consulta el estado actual del perfil a través de la interfaz con ISD-R.

2. Si se requiere una activación o cambio de perfil:

   - Se comunica con ISD-P mediante comandos APDU.
   - Se inicia una transacción y se notifica al backend a través del API REST (método POST /eventos).

3. El backend confirma recepción y registra el evento en la blockchain mediante smart contracts.
4. Una vez confirmado el cambio, el IPA actualiza el estado interno del dispositivo.
5. En escenarios más complejos, el IPA puede actuar como cliente del eIM para coordinar descargas de nuevos perfiles desde la SM-DP+.

A continuación, en la figura 4.3 se detalla el flujo descrito en el siguiente diagrama de secuencia.

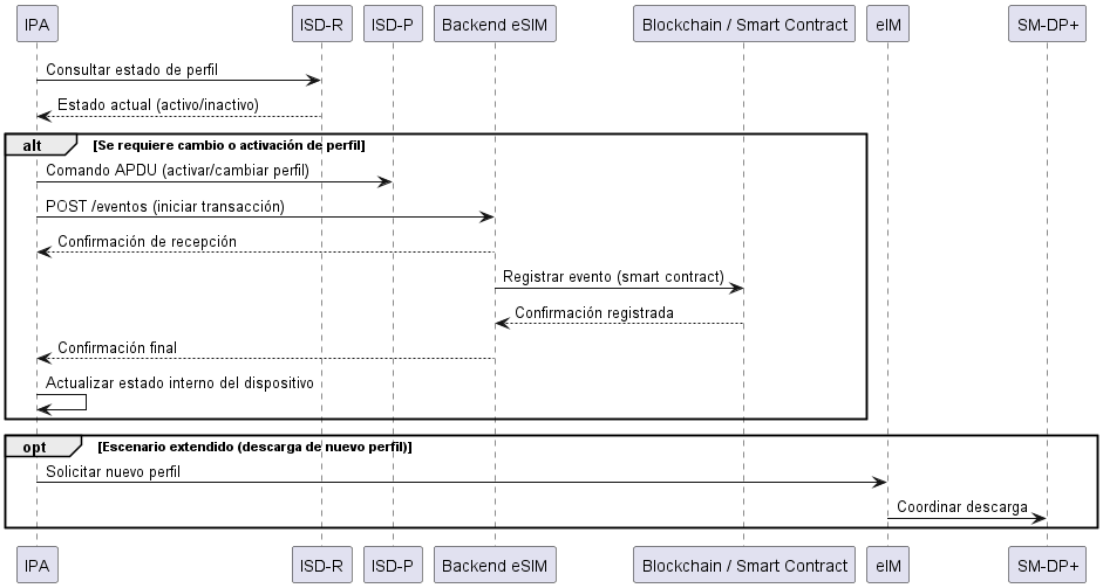

Figura 4.3. Flujo de comunicación, bloque IoT Device.

6. Posibles Extensiones

El IPA podría implementar mecanismos para monitorizar calidad de servicio (QoS) y tomar decisiones automáticas de cambio de perfil.

Como posible extensión funcional del IPA embebido en la eUICC, se contempla la integración con protocolos ligeros de comunicación ampliamente

adoptados en entornos IoT, como **MQTT** y **CoAP**. Ambos protocolos permiten una comunicación eficiente entre dispositivos de baja potencia y plataformas de gestión o monitorización remotas.

- **MQTT** (Message Queuing Telemetry Transport) es un protocolo de mensajería basado en el modelo *publish/subscribe*, optimizado para redes con alta latencia o ancho de banda limitado. Su estructura ligera y su soporte para sesiones persistentes lo hacen ideal para dispositivos embebidos. Su estándar está mantenido por OASIS [43].
- **CoAP** (Constrained Application Protocol), definido por el IETF, es un protocolo RESTful similar a HTTP pero adaptado a dispositivos y redes restringidas. Opera sobre UDP y permite interacción mediante métodos como GET, POST, PUT y DELETE, siendo ideal para arquitecturas orientadas a eventos y comunicaciones rápidas [44].

Ambos protocolos podrían utilizarse para notificar al backend eSIM o a plataformas como el eIM, eventos relevantes del IPA, como cambios de estado de perfiles, resultados de operaciones o condiciones de red detectadas, facilitando una interoperabilidad eficiente en entornos industriales y distribuidos.

4.1.1.2. eIM (eSIM IoT Remote Manager)

En la siguiente figura se muestra el eIM componente que analizaremos a continuación:

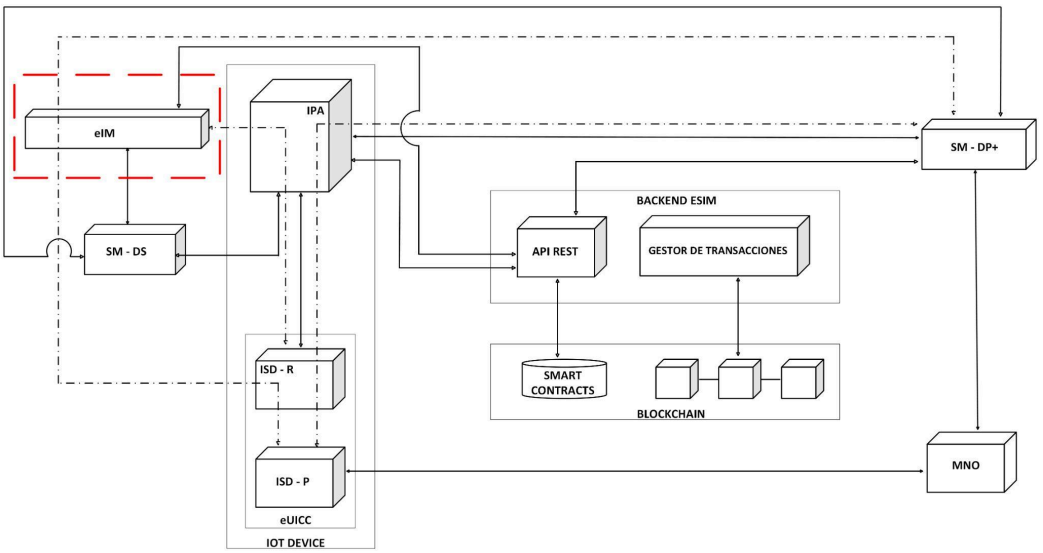

Figura 4.4. Bloque eIM (eSIM IoT Remoto Manger).

1. **Estructura Interna del Bloque**

El eIM (ver figura 4.4) es una entidad lógica definida en el estándar GSMA SGP.31/SGP.32 que actúa como intermediario entre el IPA (IoT Profile Assistant), el SM-DP+ (Subscription Manager Data Preparation+) y otros componentes del ecosistema eSIM. Su función principal es facilitar la gestión remota de perfiles eSIM en dispositivos IoT, especialmente en despliegues a gran escala.

2. **Componentes internos:**

- Módulo de gestión de perfiles: Encargado de enviar comandos de gestión de estado de perfiles (activación, desactivación, eliminación) a los dispositivos IoT.
- Interfaz de comunicación con IPA: Permite la interacción con el IPA para coordinar operaciones de gestión de perfiles.
- Interfaz de comunicación con SM-DP+: Facilita la descarga y provisión de perfiles desde el SM-DP+ hacia los dispositivos IoT.
- Módulo de seguridad: Gestiona la autenticación, autorización y establecimiento de canales seguros con los dispositivos y otros componentes.

3. **Funciones del eIM**

- Gestión de estado de perfiles: Envía comandos para activar, desactivar o eliminar perfiles en los dispositivos IoT.
- Coordinación de descargas de perfiles: Orquesta la descarga de perfiles desde el SM-DP+ hacia los dispositivos IoT, asegurando que se realice de manera segura y eficiente.
- Interacción con el IPA: Se comunica con el IPA para coordinar operaciones de gestión de perfiles y recibir notificaciones de eventos relevantes.
- Gestión de flotas de dispositivos: Facilita la gestión masiva de dispositivos IoT, permitiendo operaciones en lote y automatización de tareas comunes.

4. **Protocolos**

- CoAP (Constrained Application Protocol) sobre UDP: Utilizado para comunicaciones ligeras y eficientes con dispositivos IoT, especialmente en redes de baja potencia y ancho de banda limitado.

- HTTPS sobre TCP/IP: Empleado para comunicaciones más robustas y seguras con componentes como el SM-DP+.
- DTLS (Datagram Transport Layer Security): Proporciona seguridad en las comunicaciones basadas en UDP, asegurando la confidencialidad e integridad de los datos.
- ASN.1 y JSON: Formatos de codificación de datos utilizados para estructurar la información intercambiada entre componentes.

5. Mecanismos de Seguridad

- Autenticación mutua: Establecimiento de canales seguros mediante certificados digitales y autenticación mutua entre el eIM y los dispositivos IoT.
- Autorización basada en Roles: Control de acceso a funciones y datos según los roles asignados a cada entidad.
- Cifrado de datos: Protección de la información en tránsito mediante cifrado, asegurando la confidencialidad y la integridad de los datos.
- Gestión de claves: Administración segura de claves criptográficas utilizadas en las comunicaciones y operaciones de gestión de perfiles.

6. Flujo de Comunicación

El flujo de comunicación del eIM (eSIM IoT Remote Manager) se basa en su rol central de orquestación en la gestión remota de perfiles eSIM. Su interacción no se limita al dispositivo IoT, sino que se extiende al Backend eSIM y a la infraestructura del proveedor de servicios (SM-DP+), permitiendo una coordinación segura y trazable. A continuación se describe el flujo detallado:

1. El eIM recibe una solicitud de operación desde un sistema de gestión (OSS/BSS) o una aplicación externa, indicando una acción a realizar sobre un perfil eSIM (por ejemplo, activación, eliminación, cambio de operador).
2. El eIM establece una comunicación con el IPA embebido en el dispositivo IoT, consultando el estado actual de la eUICC, disponibilidad de perfiles, información de conectividad, y verificando si el dispositivo está en condiciones de ejecutar la operación.
3. De forma paralela, el eIM invoca el API REST del Backend eSIM para:

    - Registrar la solicitud de operación y crear una transacción identificable.

- Consultar reglas de negocio, atributos asociados al perfil o políticas de validación.
- Activar la lógica de trazabilidad: el Backend genera y envía el evento correspondiente para ser almacenado en la blockchain mediante un smart contract.
- Obtener, si corresponde, información técnica asociada al perfil (por ejemplo, operador asignado, restricciones de uso, duración, QoS esperado).

4. Si la operación requiere una descarga de perfil, el eIM actúa como intermediario entre el SM-DP+ y el dispositivo IoT, coordinando la operación a través del SM-DS (Secure Routing Server) cuando es necesario.
5. Una vez descargado el perfil o validada la operación, el eIM instruye al IPA para proceder con la activación, desactivación o eliminación del perfil en la eUICC, utilizando protocolos seguros como HTTPS/TLS 1.3, CoAP/DTLS o interfaces definidas por el estándar GSMA SGP.31/32.
6. El IPA envía una respuesta al eIM informando el resultado de la operación.
7. El eIM actualiza al backend eSIM a través de su API, permitiendo que:

    - El evento final sea almacenado en la blockchain como parte de la trazabilidad.
    - Se actualicen los registros y dashboards administrativos.
    - Se mantenga un historial completo del ciclo de vida del perfil.

A continuación, en la figura 4.5 se muestra de manera más detallada el flujo descrito:

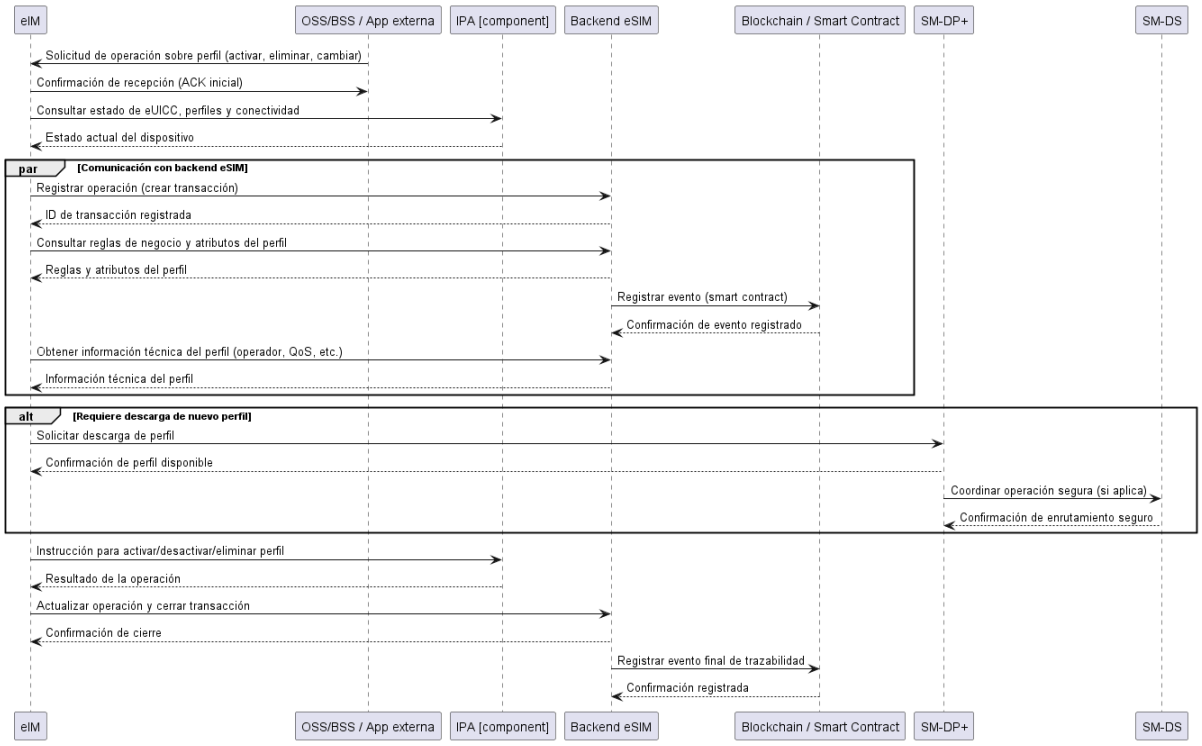

Figura 4.5. Flujo de comunicación, bloque eIM.

4.1.1.3. Backend eSIM

En la siguiente figura se muestra el Backend eSIM componente que analizaremos a continuación:

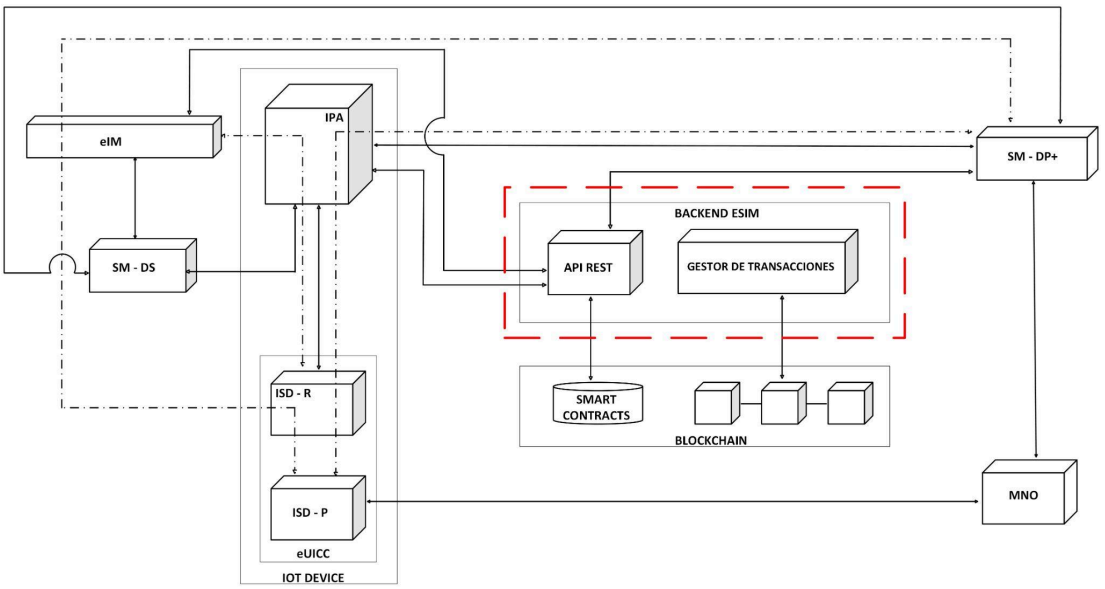

Figura 4.6. Bloque backend eSIM.

1. **Arquitectura Interna del Bloque**

El Backend eSIM (ver figura 4.6) es una estructura que implementa la lógica de negocio necesaria para orquestar, validar y auditar operaciones sobre perfiles eSIM, integrando además un módulo de interacción con la blockchain. Su arquitectura modular permite desacoplar funciones clave y escalar de forma independiente los distintos servicios.

2. **Componentes principales:**

- API REST gateway: Punto de entrada para interacciones externas: IPA, eIM, interfaces de administración, etc.
- Motor de Lógica de operación: Encargado de validar operaciones, aplicar políticas y transformar datos.
- Módulo de eventos blockchain: Responsable de instanciar y ejecutar smart contracts en una red blockchain (Polkadot/Ethereum, etc).
- Base de Datos Local (NoSQL/SQL): Registro auxiliar y persistente de los eventos, estado de las operaciones y dispositivos.
- Servicios de Integración: Adaptadores para comunicar con SM-DP+, SM-DS u otros entornos externos (cuando aplique).
- Panel de Administración / API para dashboards: Permite a operadores, prestadores de servicio y clientes,visualizar el estado de los dispositivos, historial de operaciones, fallos, métricas, etc.
- Como componente principal de la arquitectura se puede implementar en un entorno clusterizado y en alta disponibilidad, el cual aporta redundancia y robustez a nivel de infraestructura.

3. **Endpoints API REST**

A continuación, en la tabla 4.1 se listan los principales endpoints que se podrían implementar:

Ruta	Método	Descripción
/devices	GET	Lista todos los dispositivos registrados o uno por eID
/register	POST	Registra un nuevo dispositivo (sin identidad blockchain)

/register_identity	POST	Registra un dispositivo con identidad blockchain
/change_operator	POST	Cambia el operador actual de un dispositivo
/operator_history/#eID	GET	Historial de cambios de operador
/device_identity	GET	Consulta la clave pública del dispositivo desde smart contract
/auth_challenge	POST	Genera un nonce para autenticación con firma digital
/auth_verify	POST	Verifica una firma digital sobre el nonce
/blockchain/:tx_hash	GET	Consulta estado de una transacción en blockchain
/events	GET	Lista eventos on-chain de perfiles eSIM
/events	POST	Registra un evento firmado de gestión de perfil
/profile_status	GET	Consulta estado del perfil por ICCID (Integrated Circuit Card Identifier)
/validate_profile	POST	Simula validación técnica de un perfil
/audit_logs	GET	Registros de auditoría técnica del backend
/device_stats	GET	Métricas agregadas de uso y actividad
/search_devices	GET	Busca dispositivos con base en diversos filtros
/device_details/:id	GET	Obtiene información detallada de un dispositivo
/device_details/:id	PUT	Modifica atributos del dispositivo (plan, SIM, etc)
/device_location/:id	GET	Historial de ubicación por período de tiempo
/device_usage/:id	GET	Uso general del dispositivo
/device_usage_zone/:id	GET	Uso por zona y ciclo de facturación
/device_session/:id	GET	Detalles de la sesión más reciente del dispositivo

Tabla 4.1. Endpoints API REST.

4. Validaciones y lógica de operación

1. El Backend aplica múltiples validaciones antes de autorizar o registrar una operación:

2. Verificación del origen del evento (firma JWT, IP autorizada, timestamp válido).
3. Validación de estado del dispositivo y del perfil (por ejemplo: no activar si ya está activo, no cambiar de operador si ya se encuentra registrado en dicho operador, etc).
4. Autorización según reglas de operador: por ejemplo, si un MNO tiene exclusividad o ventanas de uso, plan de servicio exclusivo o con restricciones.
5. Tolerancia a fallos y reintentos seguros (timeouts, replays).

Además, podría implementar reglas para gobernanza distribuida, como:

- Control de que una sola identidad activa esté registrada por dispositivo.
- Chequeo de consumo, QoS o roaming previo a cambiar de operador (integración opcional con plataformas MNO).

5. Mecanismos de Seguridad

- TLS 1.3 con autenticación para todas las conexiones.
- Tokens firmados (JWT RS256) con roles y expiración.
- Hash y firma digital de los datos registrados en blockchain.
- Auditoría continua de acceso a logs y eventos (inmutable).

Se plantea la posibildad de implementar códigos de autenticación de mensajes basados en hash (HMAC) como mecanismo adicional de integridad para eventos sensibles, como la activación de perfiles o cambios de estado críticos. HMAC permite garantizar que el contenido del mensaje no ha sido alterado y que proviene de una fuente autenticada, utilizando una combinación de una clave secreta y una función hash criptográfica. Esta técnica es ampliamente utilizada en protocolos seguros y está estandarizada en el [RFC 2104][45].

6. Integración con Blockchain

El Backend se conecta a una blockchain, en la que se despliegan smart contracts para registrar operaciones como:

- Activación o cambio de perfil.
- Asignación o cambio de operador.
- Trazabilidad de estado de conectividad.

- Cambios de estado
- Verificación de identidad

7. **Cada evento registrado incluye:**

- Descripción
- ID de la transacción
- Estado
- Hash
- ID del dispositivo

Para entender un poco más el flujo de este componente tan importante dentro la arquitectura propuesta, se muestra a continuación el siguiente diagrama (ver figura 4.7).

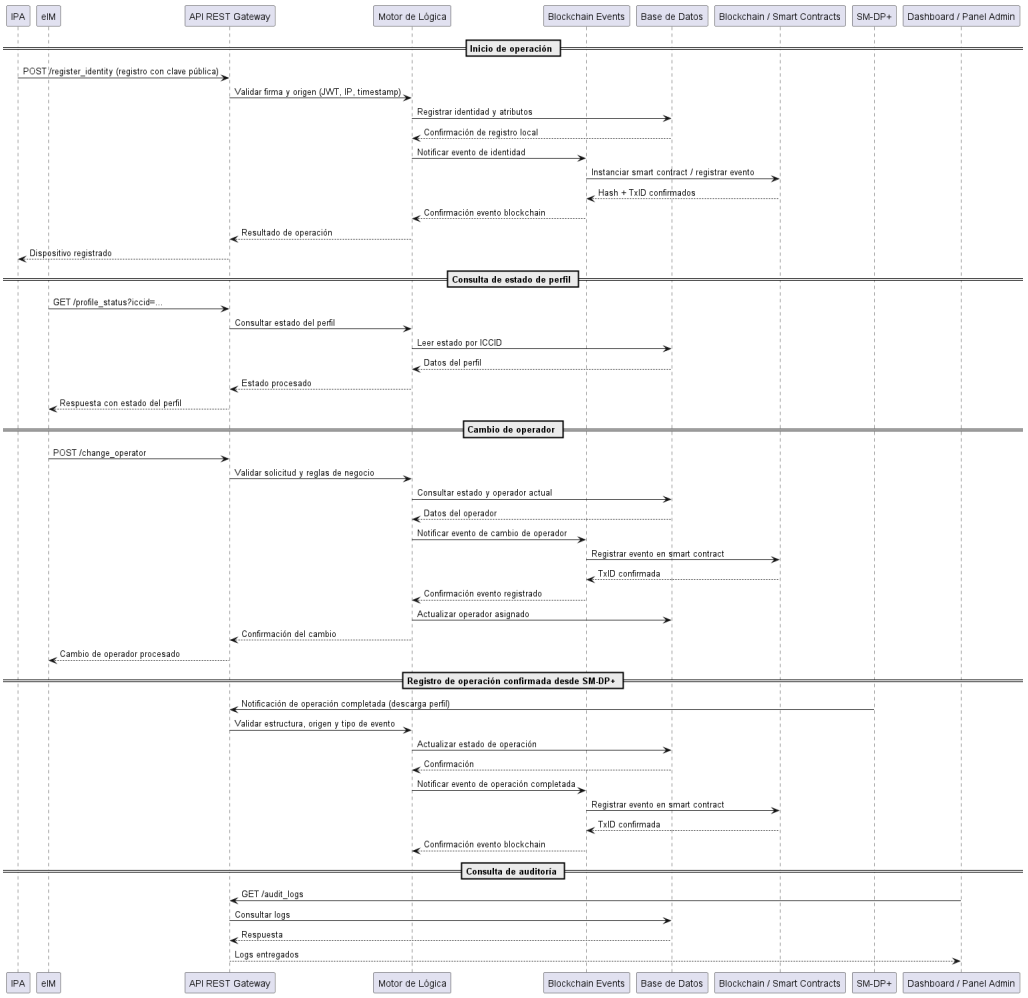

Figura 4.7. Flujo de comunicación, Backend eSIM.

4.1.1.4. SM-DP+ (Subscription Manager - Data Preparation+)

El SM-DP+ es una entidad lógica fundamental dentro del ecosistema eSIM definida por los estándares de la GSMA (SGP.21/22 y SGP.31/32). Se implementa generalmente como una plataforma de software de propósito específico, alojada en una infraestructura segura y auditada, la cual puede estar desplegada sobre servidores físicos dedicados o entornos virtualizados en centros de datos certificados (Tier III o superiores). No se trata de un componente hardware en el dispositivo ni embebido, sino de una infraestructura backend mantenida por el MNO (Mobile Network Operator) o por proveedores certificados por la GSMA (entidades SM-DP+ as-a-Service).

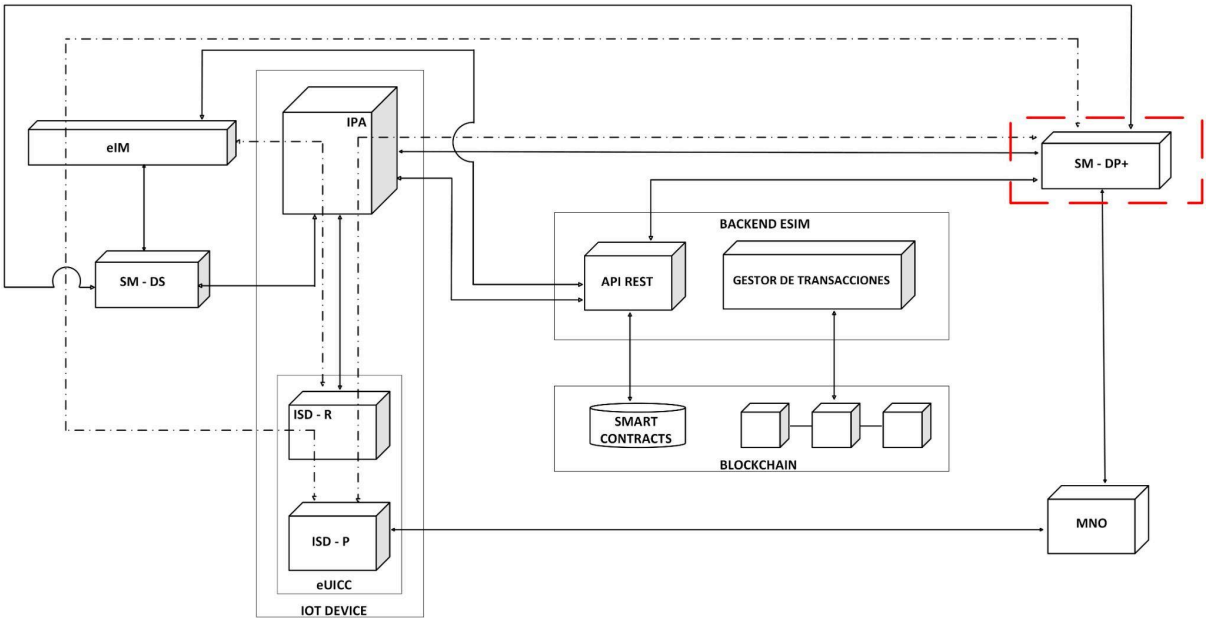

Figura 4.8. Bloque SM-DP+

Este módulo (ver figura 4.8) está compuesto internamente por varios subcomponentes software, encapsulados dentro de su estructura:

- Módulo de cifrado y empaquetado de perfiles: Encargado de preparar los perfiles de operador (con claves, identificadores IMSI, algoritmos de autenticación, etc.) en un formato seguro siguiendo el estándar de codificación ASN.1/DER.
- Módulo de gestión de identidades (eIDAS/PKI): Administra los certificados digitales y realiza la firma criptográfica de los perfiles con claves privadas del MNO o autoridad certificadora autorizada.

- Repositorio seguro de perfiles: Base de datos cifrada que almacena los perfiles en estado preparado, listos para ser descargados una vez autorizados.
- API de provisión y autorización: interfaz de comunicación autenticada con el SM-DS (Secure Routing Server) o directamente con el dispositivo en arquitecturas simplificadas (como en SGP.32).
- Módulo de logging y auditoría: Registra todas las operaciones de generación, exportación, entrega e instalación de perfiles en logs inmutables y auditables.
- Entorno HSM (Hardware Security Module) o equivalente: Utilizado para proteger claves privadas y ejecutar operaciones criptográficas dentro de un perímetro físicamente seguro y certificado (FIPS 140-2 nivel 3 o superior).

El SM-DP+ es una entidad centralizada gestionada por el MNO cuando se trata de operadores que poseen su propia infraestructura eSIM, o por un proveedor externo certificado cuando los MNO delegan este servicio. Es responsabilidad del MNO garantizar la disponibilidad, integridad, y trazabilidad de este sistema, ya que es el único autorizado para emitir y distribuir perfiles operativos válidos para ser instalados en eUICCs.

1. **Funciones Principales**

    - Almacenamiento de Perfiles: Mantiene una base de datos segura de perfiles eSIM encriptados, listos para ser entregados a dispositivos autorizados.
    - Cifrado y Personalización: Aplica procesos de cifrado y personalización de los perfiles con claves específicas por dispositivo, garantizando la confidencialidad del contenido.
    - Entrega Segura de Perfiles: Transmite los perfiles al dispositivo destino utilizando canales autenticados, ya sea a través del SM-DS o mediante enlace directo con el eIM.
    - Gestión de Estado de Perfil: Permite operaciones como activación, suspensión, eliminación o transferencia de perfiles hacia nuevos dispositivos.

2. **Protocolos y Seguridad**

    - TLS 1.2/1.3: Para el cifrado de todas las comunicaciones.

- PKI (Infraestructura de Clave Pública): Para la autenticación mutua entre SM-DP+, eIM y dispositivos.
- APDU over HTTPS: Transmisión de comandos de gestión de perfiles encapsulados sobre HTTPS.
- GSMA ES9+ y ES10b: Interfaces estandarizadas para interoperabilidad con componentes como SM-DS, eIM e IPA.

3. Flujo de Comunicación

1. El eIM solicita al SM-DP+ el envío de un nuevo perfil a un dispositivo IoT autorizado.
2. El SM-DP+ cifra el perfil utilizando las claves públicas del eUICC del dispositivo destino.
3. El perfil se transmite al dispositivo, ya sea directamente o a través del SM-DS (Secure Routing Server).
4. Una vez recibido, el perfil es instalado y activado por el IPA en el dispositivo, bajo supervisión del eIM.
5. El SM-DP+ actualiza el estado del perfil en su sistema y confirma al eIM y al backend eSIM el éxito de la operación, para que posteriormente sea registrado en la blockchain. Ver flujo de comunicación en la figura 4.9.

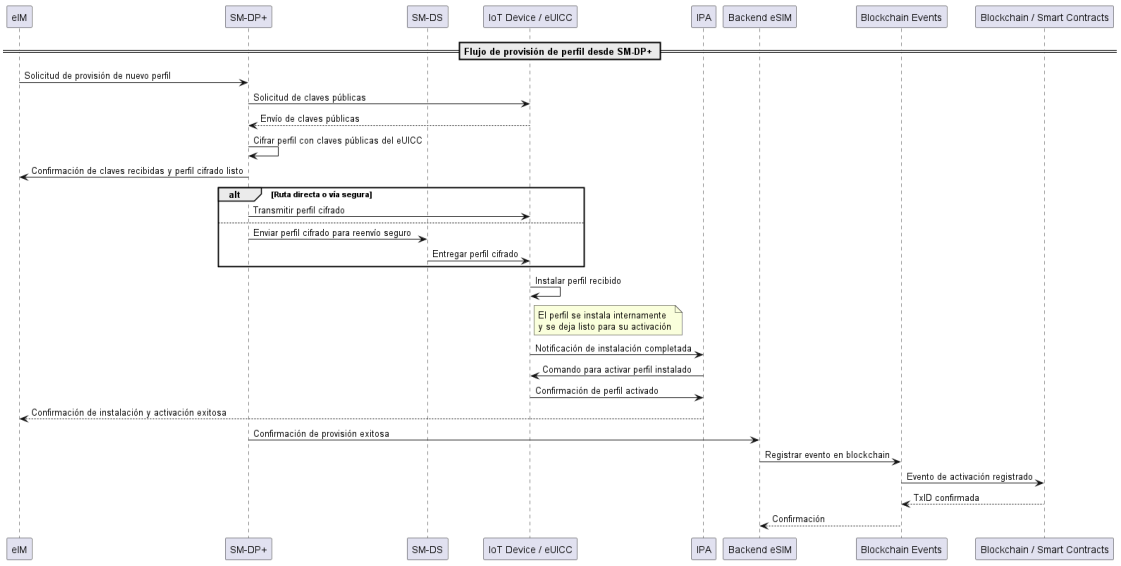

Figura 4.9. Flujo de comunicación, SM-DP+.

4.1.1.5. Mobile Network Operator (MNO)

El MNO (ver figura 4.10) u Operador de Red Móvil, es una entidad operativa y comercial encargada de ofrecer servicios de conectividad móvil. En el contexto de la arquitectura eSIM basada en los estándares GSMA SGP.31/32, el MNO representa el actor que provee los perfiles de red que serán descargados, activados y utilizados en dispositivos IoT a través del entorno eUICC.

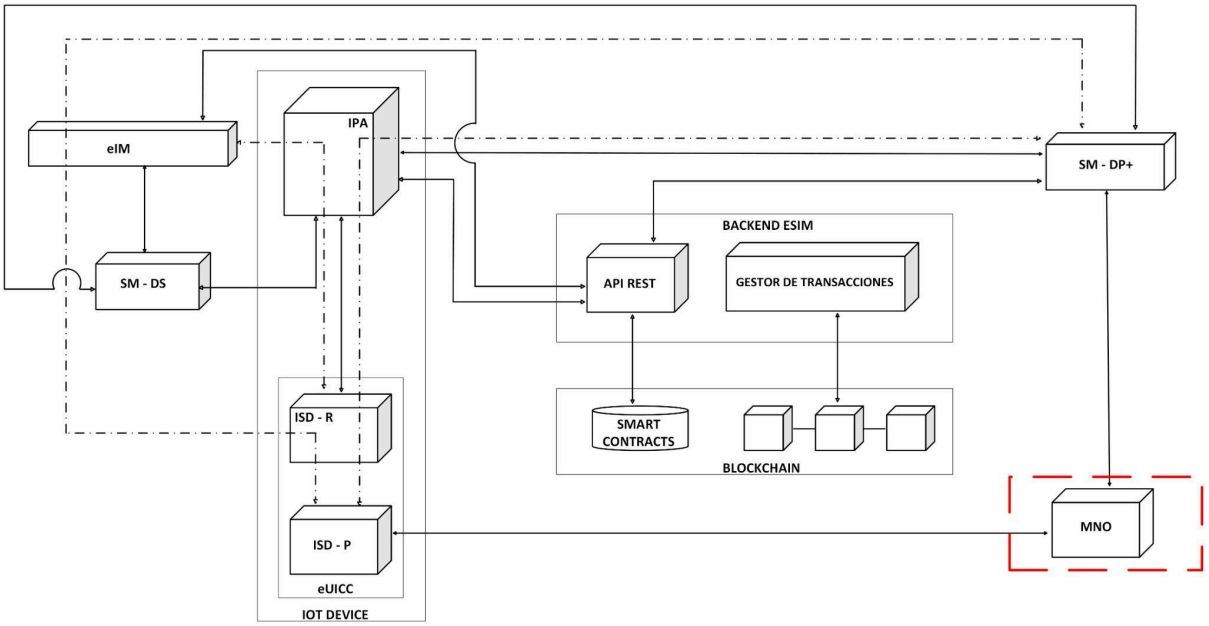

Figura 4.10. Bloque Operador Móvil de Red (MNO).

El MNO no es una entidad física dentro del dispositivo, sino una organización externa que interactúa con la infraestructura del ecosistema eSIM a través del SM-DP+, y en esta propuesta, también mediante el backend eSIM y la red blockchain.

1. **Componentes típicos del MNO en esta arquitectura:**

- HSS/HLR (Home Subscriber Server/Location Register): Contiene la información de abonado.
- Sistema OSS/BSS (Operations/Business Support System): Gestiona la provisión, facturación y análisis de eventos.
- Sistema de emisión de perfiles: Herramienta que genera y firma los perfiles eSIM, que luego son enviados al SM-DP+.

2. Función

En la arquitectura propuesta, el MNO cumple con las siguientes funciones:

- Proporciona los perfiles eSIM que contienen credenciales IMSI, claves de autenticación, y parámetros de conectividad según los servicios provisionados.
- Solicita al SM-DP+ la descarga de perfiles en los dispositivos IoT cuando corresponda, ejemplo cuando se genera el alta de un servicio solicitado.
- Interactúa con el backend eSIM de forma indirecta a través del SM-DP+ o mediante APIs cuando se requiere validar condiciones específicas de negocio/operación (como consumo local y en roaming, cobertura o cambios de operador, etc).

3. Protocolos

- HTTPS/TLS 1.3: Para la comunicación con el SM-DP+ y plataformas externas.
- GSMA SGP.22/SGP.32: Para el formato y la entrega de perfiles eSIM.
- RADIUS/Diameter: En la autenticación de dispositivos al momento de registrar en la red.
- REST/JSON APIs: Para integraciones con el backend eSIM y otras plataformas de terceros.

4. Mecanismos de Seguridad

- Emisión de perfiles firmados digitalmente mediante certificados X.509 emitidos por la CA del operador.
- Validación del perfil antes de la descarga en el dispositivo mediante mecanismos criptográficos definidos por GSMA.
- Canales cifrados y autenticación mutua (mTLS) con SM-DP+ para la entrega de perfiles.
- Auditoría de descargas y activaciones registrada internamente en el sistema OSS/BSS del MNO y, en esta arquitectura, también en la blockchain mediante el backend.

5. Flujo de Comunicación

1. El MNO genera un nuevo perfil eSIM y lo registra en su sistema de gestión.

2. Este perfil es firmado y enviado al SM-DP+, encargado de su preparación y distribución.
3. Cuando el eIM inicia una operación de descarga (por ejemplo, por decisión del IPA o por evento programado), el SM-DP+ entrega el perfil al dispositivo IoT a través del SM-DS.
4. Una vez activado el perfil, el dispositivo IoT se registra en la red del MNO utilizando las credenciales recibidas.
5. Paralelamente, el backend eSIM registra esta transacción como evento firmado en la blockchain, permitiendo trazabilidad y validación posterior.
6. El MNO puede consultar esta trazabilidad o integrarse mediante API al backend para propósitos regulatorios, facturación o gestión operativa.

El siguiente diagrama (ver figura 4.11) muestra el flujo de comunicación desde el MNO.

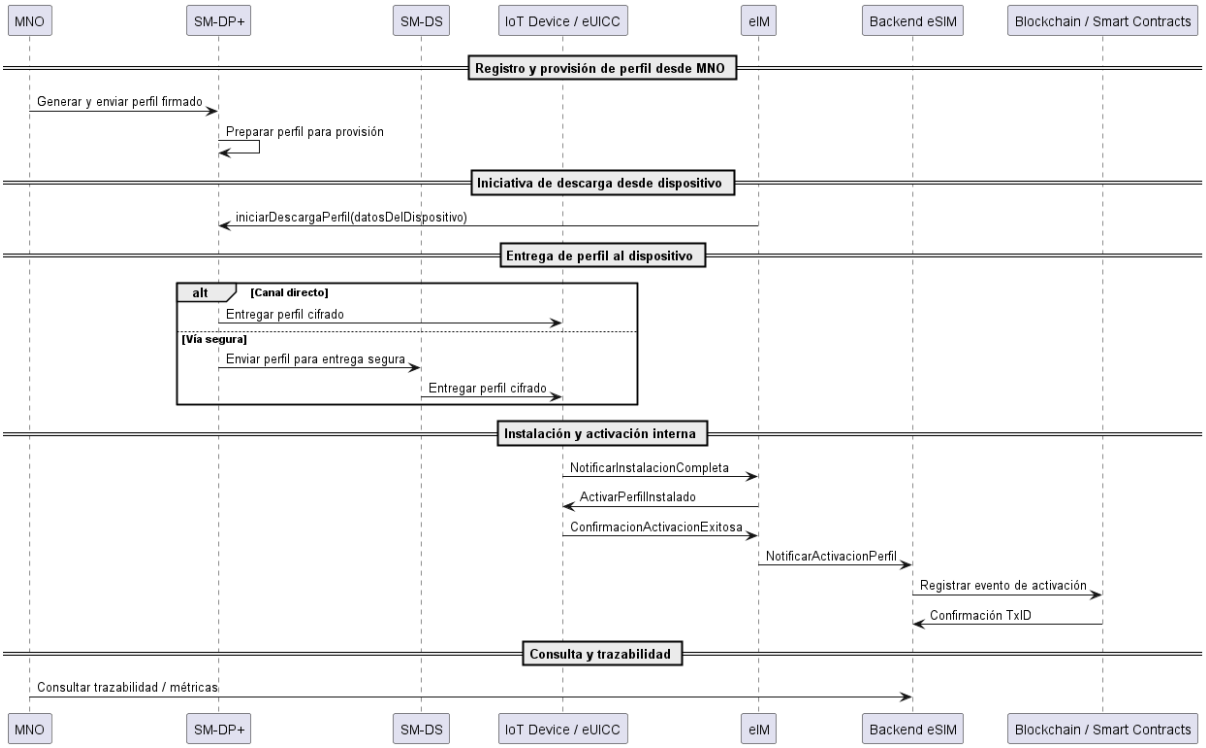

Figura 4.11. Flujo de comunicación MNO.

4.1.1.6. SM-DS (Subscription Manager – Discovery Server)

El SM-DS es una entidad lógica definida en el estándar GSMA SGP.22/SGP.32, cuyo propósito es facilitar la entrega de notificaciones a los dispositivos IoT eUICC

para indicarles la disponibilidad de nuevos perfiles remotos que deben ser descargados desde un SM-DP+ autorizado.

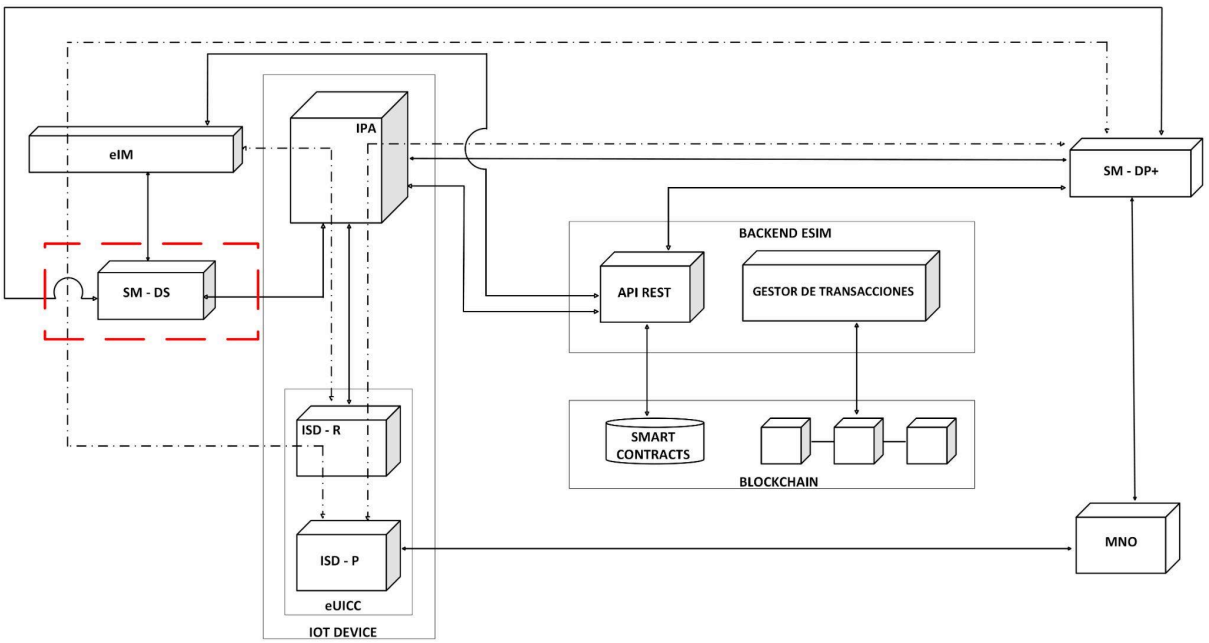

Figura 4.12. Bloque SM-DS.

El SM-DS (ver figura 4.12) es típicamente una infraestructura de software desplegada en la nube o en una red perimetral segura, mantenida por una entidad de confianza certificada, que puede ser el MNO, el fabricante de la eUICC, o un proveedor acreditado independiente. Su diseño está optimizado para garantizar escalabilidad, alta disponibilidad y compatibilidad con múltiples dispositivos conectados simultáneamente a través de redes públicas o privadas. No contiene lógica de provisión de perfiles, sino que actúa como punto de notificación y descubrimiento de URLs de descarga de perfiles.

1. **Componentes internos:**

- Servidor HTTPS: Expone una interfaz RESTful para recibir y almacenar notificaciones del SM-DP+.
- Módulo de notificación push: En algunos casos puede integrarse un mecanismo de notificación activa hacia el dispositivo.
- Almacenamiento temporal: Base de datos o almacenamiento en memoria para gestionar mensajes pendientes de consulta por parte de los dispositivos.

- Registro de auditoría: Bitácora de operaciones para trazabilidad de mensajes y consultas.

2. Función

El SM-DS permite a los dispositivos descubrir automáticamente que hay perfiles disponibles para descarga, eliminando la necesidad de configuración manual o supervisión directa por parte del usuario o administrador. Funciones clave:

- Recibir notificaciones del SM-DP+ cuando hay perfiles nuevos disponibles.
- Asociar notificaciones a un EID.
- Servir como endpoint de consulta para que los dispositivos IoT verifiquen si hay perfiles pendientes.
- Facilitar la redirección segura hacia el SM-DP+ responsable del aprovisionamiento.

3. Protocolos

- HTTPS sobre TLS 1.2/1.3: Canal seguro entre SM-DP+ y SM-DS, y entre SM-DS y eUICC.
- JSON-RPC / RESTful API: Intercambio de datos estructurados (notificaciones, peticiones, respuestas).
- EID-Based Lookup: Identificación del dispositivo IoT basada en su identificador EID.
- GSMA SGP.22 / SGP.32: Estándares que definen el formato de mensajes, URLs y comportamiento esperado.

4. Mecanismos de Seguridad

- Autenticación mutua TLS: Validación de identidad tanto del cliente (SM-DP+) como del servidor (SM-DS).
- Control de acceso basado en roles (RBAC): Asegura que solo componentes autorizados puedan registrar o consultar notificaciones.
- Cifrado en tránsito y en reposo: Todos los datos, incluyendo las URLs de provisión, se cifran con algoritmos AES-256 o equivalentes.
- Recolección de logs firmados digitalmente: Para auditoría de eventos e integridad de operaciones.

- Validez temporal de las notificaciones: Las entradas tienen TTL (time-to-live) configurables para evitar uso indebido.

5. Flujo de Comunicación

1. El SM-DP+, tras generar un nuevo perfil eSIM, registra una URL de provisión segura en el SM-DS, junto con el EID del dispositivo destinatario.
2. El SM-DS almacena esta información temporalmente, junto con una marca de tiempo y, opcionalmente, una clave de acceso o token asociado.
3. El dispositivo IoT (a través del IPA) inicia una consulta al SM-DS usando su EID y recibe como respuesta una lista de URLs disponibles.
4. El IPA válida la URL y se comunica directamente con el SM-DP+ para iniciar la descarga segura del perfil eSIM.
5. Una vez descargado el perfil, el dispositivo puede borrar la entrada o esperar su expiración automática. Ver flujo de comunicación en la Figura 4.13

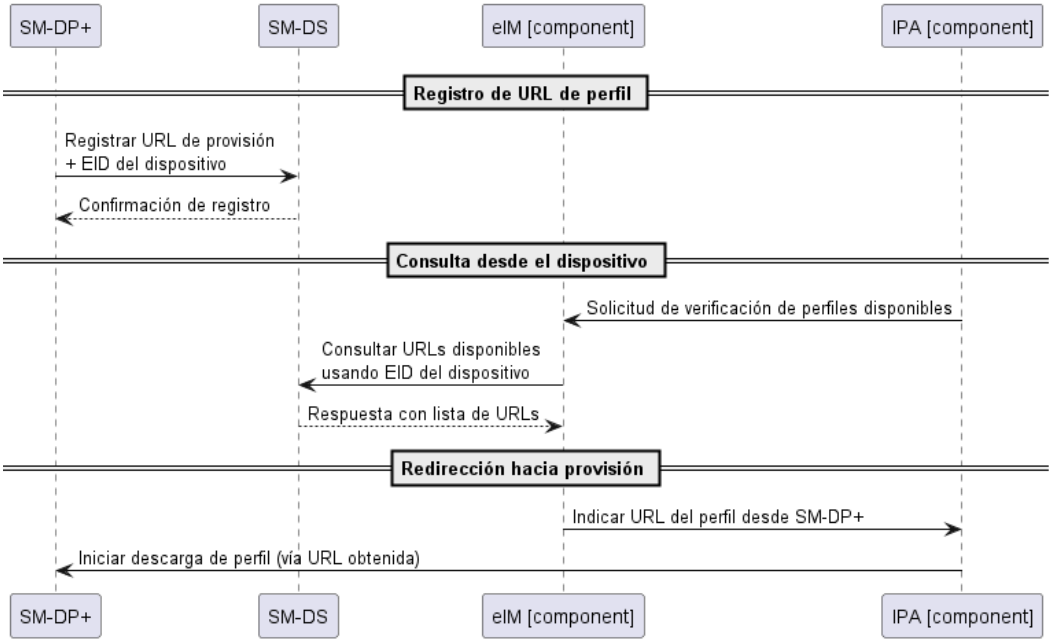

Figura 4.13. Flujo de comunicación SM-DS.

4.1.1.7. Blockchain

En la siguiente figura se muestra la Blockchain componente que analizaremos a continuación:

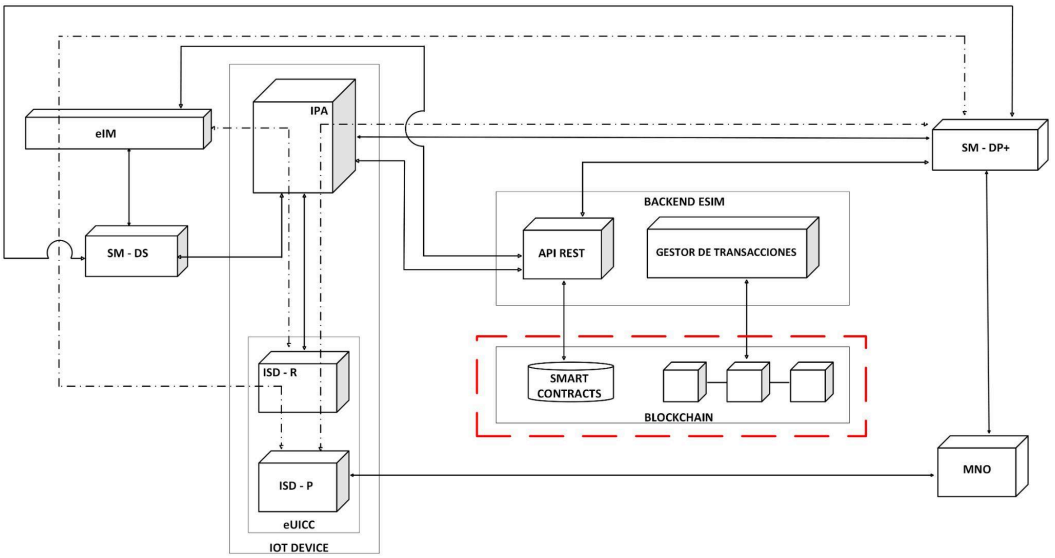

Figura 4.14. Bloque blockchain.

1. Estructura interna del bloque

El bloque de Blockchain (ver figura 4.14) representa la capa distribuida e inmutable de almacenamiento y validación de eventos críticos relacionados con la gestión de perfiles y eventos de eSIM en dispositivos IoT. En esta arquitectura, la blockchain funciona como un libro mayor compartido donde se registran operaciones verificadas desde el backend eSIM, como activaciones, cambios de operador, y eventos de red relevantes.

Este bloque está compuesto por:

- Smart Contracts: Lógica programática desplegada en la blockchain para registrar y validar eventos específicos, como la activación de un perfil eSIM o el registro de una transacción de aprovisionamiento.
- Nodos de Red: Múltiples nodos interconectados que conforman la red blockchain.
- Motor de Consenso: Algoritmo (Proof-of-Authority o Proof-of-Stake) encargado de validar los bloques en entornos de pruebas controlados o productivos.

2. Función

El objetivo de integrar blockchain en esta arquitectura es proveer al sistema de un mecanismo seguro, auditable y descentralizado para:

- Registrar el historial de operaciones eSIM de forma inmutable.
- Proveer trazabilidad completa de las acciones sobre perfiles de red en eSIM en dispositivos IoT (activaciones, desactivaciones, cambios, etc).
- Facilitar la auditoría técnica y regulatoria de la infraestructura.
- Servir como prueba criptográfica de integridad y legitimidad de los eventos.

3. Protocolos

- **JSON-RPC:** Protocolo de llamada a procedimientos remotos (RPC) basado en JSON, utilizado como estándar para la comunicación entre el backend eSIM y los nodos de la red blockchain. Es ampliamente soportado por clientes como Geth o Hardhat y permite ejecutar funciones expuestas por los nodos, como la lectura del estado o la emisión de transacciones [46].
- **Web3.js/Ethers.js:** Bibliotecas JavaScript diseñadas para interactuar con redes blockchain compatibles con Ethereum. Web3.js es uno de los clientes oficiales de la Fundación Ethereum, mientras que Ethers.js se enfoca en la simplicidad, seguridad y compatibilidad con TypeScript. Ambas se usan en el backend para firmar, enviar y consultar transacciones [47][48].
- **Smart Contracts Solidity:** Lenguaje de programación dominante en entornos compatibles con la Ethereum Virtual Machine (EVM). Utilizado para definir la lógica y estructuras de almacenamiento en contratos inteligentes desplegados en la blockchain [49].

- **IPFS (opcional):** Sistema de archivos distribuido que permite el almacenamiento descentralizado de datos como certificados, hashes de documentos o evidencias asociadas a la gestión de perfiles. Aunque aún no está implementado, se contempla su integración futura para mejorar la resiliencia y la trazabilidad [50].

4. Mecanismos de Seguridad

- **Firmas Digitales (ECDSA):** Se emplean algoritmos criptográficos robustos para firmar transacciones y eventos registrados en la blockchain. ECDSA (Elliptic Curve Digital Signature Algorithm) es ampliamente utilizado en blockchains como Ethereum [51].
- **Control de acceso por clave pública:** El sistema emplea un modelo de autorización basado en claves públicas. Solo entidades previamente

autorizadas como el IPA, el eIM o el backend eSIM pueden firmar eventos válidos. La validez de estas firmas se verifica on-chain mediante los contratos inteligentes.

- **Validación en contrato inteligente:** Los smart contracts contienen lógica embebida que impide ataques comunes como la duplicación de eventos, replay attacks o eventos con estructura inválida. Esta validación on-chain fortalece la seguridad y asegura la consistencia de los registros.
- **Hashes SHA-256 / Keccak256:** Para garantizar la integridad de cada evento, se aplican funciones hash criptográficas sobre la carga útil antes de su registro en la blockchain. SHA-256 es un estándar ampliamente utilizado en entornos seguros [52], mientras que Keccak256 es el algoritmo de hash utilizado por Ethereum y otras redes EVM [53].

- **Control de gas o peso computacional:** Se impone un límite al coste computacional de cada operación en la blockchain (medido en gas o peso, según la red), con el fin de evitar abusos de recursos y mantener la eficiencia del sistema. Este mecanismo es clave para mitigar ataques de denegación de servicio y congestión intencional.

5. Flujo de Comunicación

1. El IPA o el eIM genera un evento (ejemplo la activación de un perfil).
2. Este evento se envía al backend eSIM mediante una llamada a POST /api/eventos.
3. El backend valida el evento, aplica lógica de negocio/operación y prepara una transacción firmada.
4. La transacción es enviada a un contrato inteligente desplegado en la blockchain.
5. El contrato registra el evento, almacena hashes, IDs, estados y marcas de tiempo.
6. El backend actualiza su base de datos interna con el ID de transacción (tx_hash) y devuelve respuesta al IPA/eIM.
7. Opcionalmente, el backend podría exponer el endpoint GET /api/blockchain/:tx_id para consultas posteriores o verificación de auditoría.

4.2. Blockchain con IPA (IoT Profile Assistant) en eUICC

Esta arquitectura plantea una evolución en la forma en que se gestionan los perfiles eSIM en dispositivos IoT, en lugar de alojar el IoT Profile Assistant (IPA) en el

propio dispositivo, se integra directamente dentro de la eUICC, junto a los componentes de seguridad ISD-R e ISD-P. Esto convierte a la eUICC en un módulo aún más autónomo, capaz de gestionar internamente tanto los perfiles como las operaciones criptográficas y las interacciones con la blockchain.

La idea principal es que todo el procesamiento relacionado con la identidad del dispositivo, la gestión del ciclo de vida de los perfiles y la trazabilidad de eventos quede contenido en un entorno seguro como lo es la eUICC. De este modo, se mejora la seguridad general del sistema y se reduce la superficie de exposición del dispositivo IoT.

En esta arquitectura (ver Figura 4.15), el backend continúa jugando un papel clave como orquestador de flujos, actuando como puente entre la eUICC, la blockchain y otros actores como el SM-DP+, el MNO o el eIM. Su misión es facilitar la interoperabilidad, validar las transacciones y mantener un registro auditable de las operaciones.

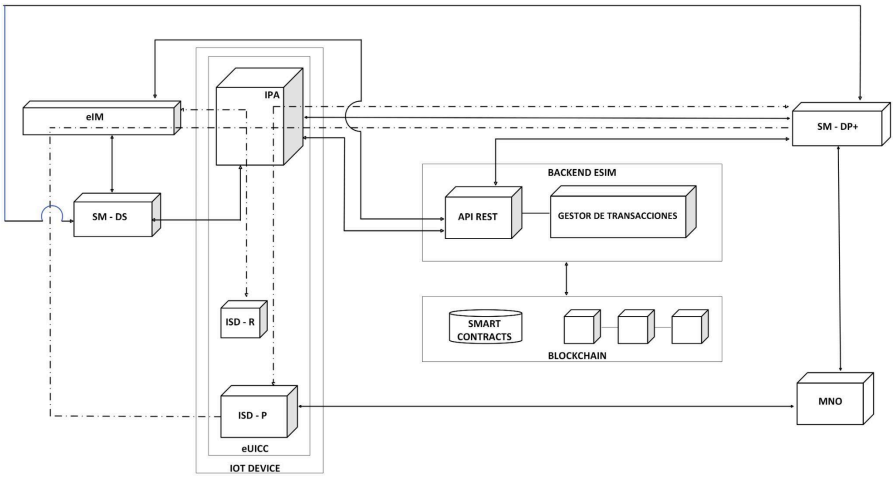

Figura 4.15. Blockchain con IPA (IoT Profile Assistant) en eUICC

A continuación, se describen cada uno de los bloques que componen esta arquitectura.

4.2.1. Componentes de la arquitectura.

4.2.1.1. eUICC

En la siguiente figura se muestra el bloque eUICC que analizaremos a continuación:

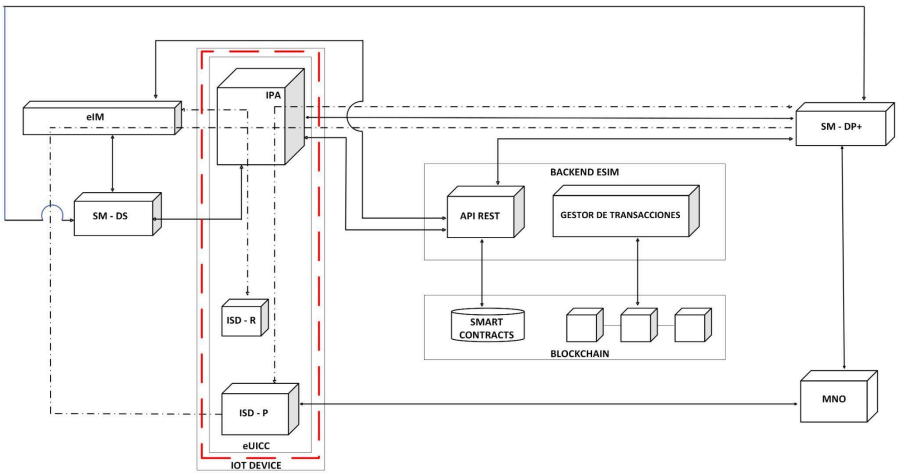

Figura 4.16. Bloque eUICC

1. **Estructura interna del bloque.**

A diferencia del modelo anterior donde el IPA estaba embebido en el dispositivo IoT, en esta arquitectura (ver figura 4.16) el componente IPA (IoT Profile Assistant) se integra directamente dentro del chip eUICC, junto a los dominios de seguridad ISD-R e ISD-P. Esta consolidación convierte a la eUICC en un módulo autónomo capaz de gestionar, autenticar y registrar transacciones de forma segura sin depender del sistema operativo del dispositivo IoT.

El ISD-R Administra y protege el acceso general a la eUICC, autorizando operaciones y gestionando la instalación de otros dominios; mientras que el ISD-P controla el ciclo de vida de los perfiles eSIM (instalación, activación, eliminación). Por otra parte el IPA se comunica con el backend y la blockchain para firmar nonces, registrar eventos relevantes (como cambios de operador), y verificar identidades. Esta arquitectura facilita la portabilidad y la seguridad, ya que el procesamiento crítico ocurre en un entorno confiable, con menor exposición a posibles vulnerabilidades externas.

2. **Función**

La eUICC se convierte aquí en el nodo principal de identidad y control en el ecosistema IoT. Su función principal es gestionar de forma segura los perfiles eSIM y autenticar operaciones mediante el IPA embebido. Gracias a su integración con la

blockchain a través del backend, cada acción crítica queda registrada con trazabilidad y garantía de integridad.

Además, el IPA embebido permite realizar operaciones como:

- Firma digital de desafíos (nonces) para autenticación.
- Validación de eventos y condiciones para activar o desactivar perfiles.
- Registro de cambios operacionales en la red blockchain.

3. Mecanismos de seguridad

Se mantienen los mecanismos de seguridad ya descritos en la arquitectura anterior:

- Almacenamiento interno cifrado.
- Generación y custodia de claves privadas dentro del entorno seguro del chip.
- Firma de mensajes mediante el módulo criptográfico de la eUICC.
- Verificación de integridad mediante hashes y autenticación mutua con el backend.
- Al estar todo contenido en el chip, se refuerza la resistencia ante ataques físicos y se reducen las posibilidades de extracción de datos sensibles.

4. Protocolos y flujo de comunicación

El flujo general de comunicación sigue la misma lógica ya descrita en la arquitectura anterior (IPA en el dispositivo IoT), con la diferencia clave de que ahora el IPA se comunica directamente desde dentro de la eUICC, a través de comandos del tipo APDU (Application Protocol Data Unit) enviados desde el sistema operativo del dispositivo o de manera remota vía OTA (Over-the-Air).

En resumen, el flujo se puede representar de esta forma:

1. El sistema operativo del dispositivo inicia la autenticación solicitando al IPA en la eUICC que firme un nonce.
2. El IPA firma el nonce con su clave privada almacenada internamente.
3. La firma es enviada al backend para ser validada y registrada en blockchain si corresponde.

4. Cualquier evento importante (como el cambio de operador o actualización de perfil) sigue un proceso similar, con el IPA firmando y validando en origen.

4.2.1.2. Backend eSIM

En la siguiente figura se muestra el el backend eSIM, componente que analizaremos a continuación:

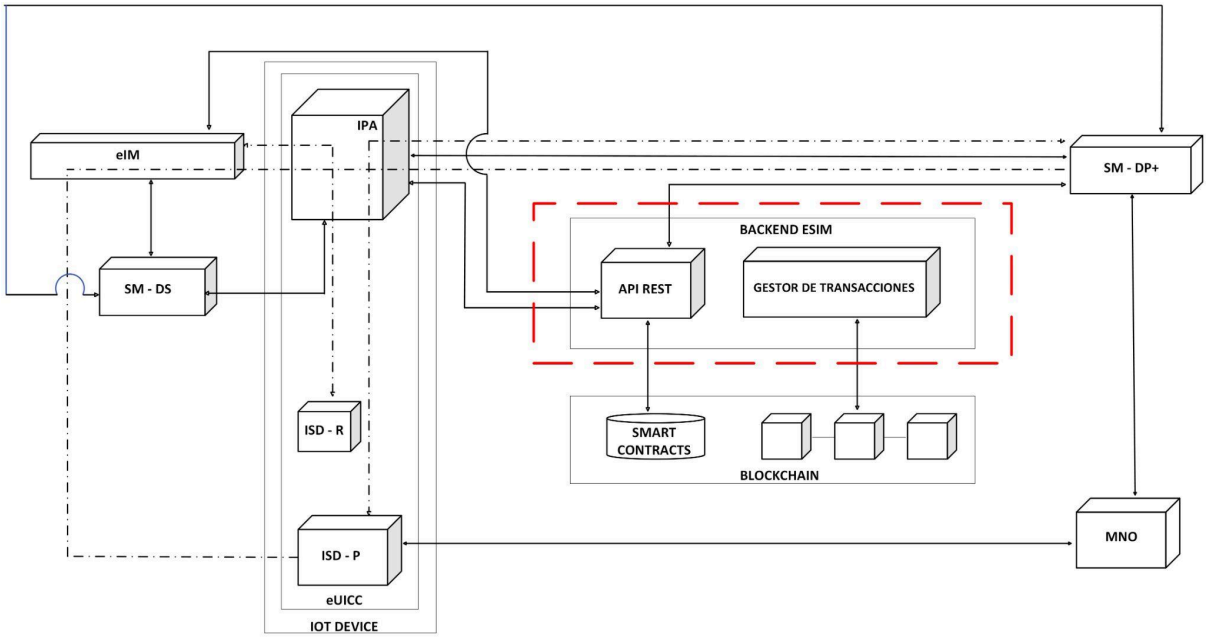

Figura 4.17. Bloque backend eSIM

1. **Arquitectura interna del bloque**

El backend eSIM (ver figura 4.17) actúa como punto central de orquestación de las operaciones relacionadas con la identidad, la autenticación y la gestión de eventos de la eUICC. Está compuesto por:

- Un módulo de API RESTful, que expone los endpoints de autenticación (/auth_challenge, /auth_verify) y registro (/register_identity, /log_event, etc.).
- Un módulo de validación y firma, que verifica la autenticidad de los mensajes firmados por el IPA.
- Un módulo de integración blockchain, que se encarga de construir, firmar y enviar transacciones hacia la red descentralizada.

- Un repositorio o base de datos para el almacenamiento temporal de nonces, identidades e historiales de eventos antes de su consolidación en blockchain.
- Como componente principal de la arquitectura se puede implementar en un entorno clusterizado y en alta disponibilidad, el cual aportaría redundancia y robustez a nivel de infraestructura.

2. Función

El backend eSIM tiene como principal función coordinar las interacciones entre el IPA embebido en la eUICC y la red blockchain, así como validar y registrar los eventos relacionados con la identidad y el ciclo de vida del perfil.

Concretamente, cumple con las siguientes responsabilidades:

- Generar y asociar un nonce único por dispositivo para cada sesión de autenticación.
- Verificar la firma digital enviada por la eUICC a través del IPA, recuperando la clave pública asociada a la identidad.
- Confirmar la correspondencia entre la dirección pública y el eID del dispositivo.
- Registrar en blockchain eventos como el aprovisionamiento de identidades, cambios de operador, activaciones, desactivaciones, etc.
- Servir como punto de integración para flujos provenientes de SM-DP+, IPA o eIM, tanto de forma individual como masiva.

3. Mecanismos de seguridad

El backend implementa diversos mecanismos de seguridad para proteger la integridad y autenticidad del sistema:

- Verificación estricta de la firma digital
- Validación cruzada entre la dirección pública, el ICCID y la clave registrada.
- Registro inmutable de eventos críticos en la blockchain.
- Gestión segura de claves privadas para la firma de transacciones blockchain.

- Además, los logs de auditoría pueden integrarse con sistemas externos o dashboards para monitoreo y análisis en tiempo real.

4. Protocolos y flujo de comunicación

El flujo de comunicación del backend eSIM no cambia respecto a la arquitectura anterior, pero en este caso, los mensajes de autenticación y eventos provienen del IPA embebido en la eUICC, y no desde el dispositivo IoT.

1. La eUICC (vía el SO del dispositivo o un canal OTA) solicita un nonce al backend (/auth_challenge).
2. El IPA firma el nonce y responde con la firma y su dirección pública al endpoint /auth_verify.
3. El backend verifica la firma y valida la identidad.
4. Una vez autenticado, se procede con el registro del evento correspondiente (/register_identity, /log_event, etc.).
5. El backend construye y firma una transacción que es enviada a la red blockchain.

Este backend es adaptable y modular, lo que le permite soportar múltiples fuentes de eventos (IPA, SM-DP+, eIM) y operar bajo entornos controlados o entornos de producción con distintas blockchains.

4.2.1.3. Blockchain

En la siguiente figura se muestra la blockchain, componente que analizaremos a continuación:

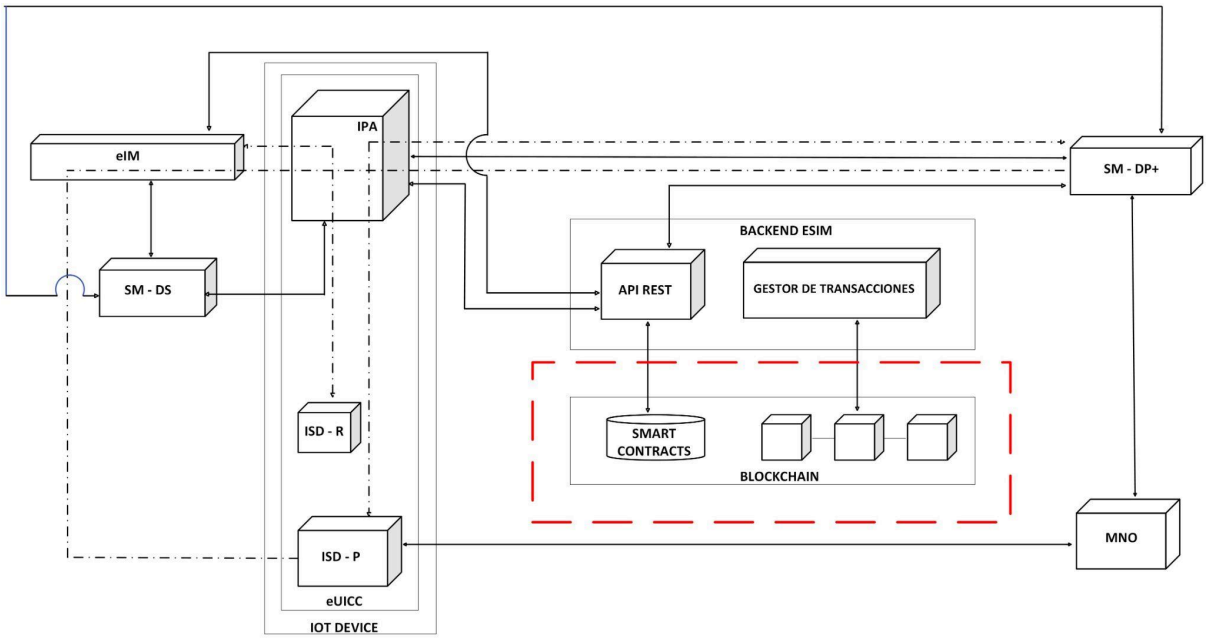

Figura 4.18. Bloque blockchain

1. Arquitectura interna del bloque

La arquitectura interna del bloque Blockchain (ver figura 4.18) permanece prácticamente idéntica a la descrita en la sección [4.1.2.7], donde se definía el uso de contratos inteligentes desplegados en redes compatibles con EVM, herramientas como Hardhat y nodos conectados por protocolo JSON-RPC.

La única diferencia significativa radica en el origen de las firmas, en este modelo, las transacciones o eventos pueden ser firmados directamente desde el IPA embebido dentro de la eUICC, sin necesidad de intermediación desde un dispositivo IoT para esta tarea en particular. Esto implica una mayor autonomía de la eUICC, que puede generar y firmar eventos críticos, como el registro de identidad o la validación de autenticación sin depender de un entorno operativo externo al chip.

2. Función

La función de la blockchain como capa descentralizada e inmutable de registro y validación de eventos sigue siendo la misma:

- Registrar identidades de eUICC asociadas a ICCID y claves públicas.

- Registrar eventos operativos relevantes (como autenticaciones exitosas, activación de perfiles, cambios de operador, etc.).
- Trazar y auditar de forma transparente todo el ciclo de vida de una eSIM, desde su emisión hasta su eventual desactivación.

Sin embargo, en esta variante, se refuerza la confianza en el origen de los eventos: al estar el IPA dentro de la eUICC, no existe intermediario que pueda alterar las firmas generadas, lo que otorga mayor robustez frente a ataques tipo man-in-the-middle o suplantación.

3. Mecanismos de seguridad

Se heredan los mecanismos detallados previamente, incluyendo:

- Firmas digitales ECDSA.
- Validación de formato y autenticidad de eventos en smart contracts.
- Control de acceso por clave pública.
- Hashing de la carga útil y control de gas.

A estos se suma un refuerzo implícito en seguridad gracias al contexto seguro que proporciona la eUICC, que no expone la clave privada del IPA a ningún software externo. Las firmas son generadas dentro del entorno aislado de la eUICC.

Protocolos y flujo de comunicación

1. Los protocolos empleados siguen siendo los mismos (JSON-RPC, Web3/Ethers.js).
2. En cuanto al flujo, el cambio clave es el origen del evento firmado
3. El evento es generado y firmado directamente dentro del componente IPA (en la eUICC).Es enviado al backend (vía POST /api/eventos) junto a la firma y el ICCID.
4. El backend válida la firma con la clave pública previamente registrada.
5. Si es válido, se construye y firma una transacción que será enviada a la red blockchain.
6. El contrato inteligente registra el evento como antes.
7. El backend responde al origen (eUICC, eIM, etc.) con el tx_hash como referencia.

Este modelo optimiza el nivel de seguridad y permite interoperabilidad multi-MNO de manera más directa, especialmente útil en esquemas donde la eUICC puede cambiar de red o perfil sin intervención del dispositivo.

4.2.2.4. SM-DP+ (Subscription Manager - Data Preparation+)

La arquitectura interna del SM-DP+ se mantiene sin cambios estructurales relevantes respecto a lo descrito en el bloque [4.1.2.4]. Sigue tratándose de una plataforma centralizada y segura, implementada por el MNO o un proveedor certificado, encargada de preparar, firmar, almacenar y entregar perfiles eSIM cifrados.

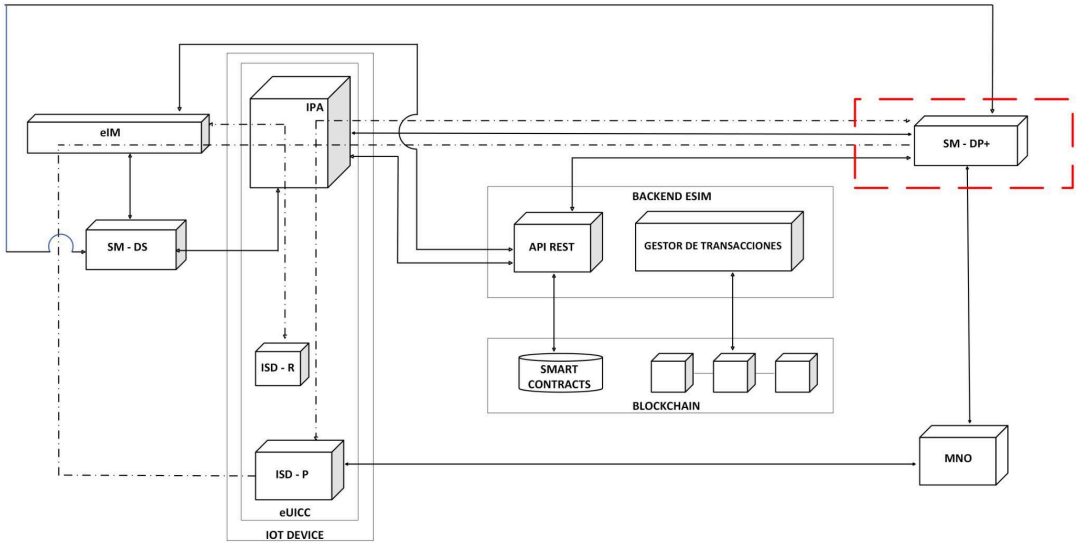

Figura 4.19. Bloque SM-DP+

Sin embargo, en esta arquitectura (ver figura 4.19) donde el IPA está embebido en la propia eUICC, se simplifica el flujo de interacción entre el SM-DP+ y el dispositivo, eliminando posibles dependencias del sistema operativo del dispositivo anfitrión. Ahora, el SM-DP+ se comunica directamente con la eUICC a través de canales autenticados, y es el IPA quien gestiona de forma autónoma la instalación del perfil desde el interior del chip. Esto permite:

- Reducir la superficie de ataque al eliminar intermediarios.
- Aumentar la estandarización del flujo de provisión, alineándose más estrechamente con el modelo SGP.32 (IoT Architecture).
- Facilitar la integración con múltiples plataformas host, ya que el control de la provisión reside por completo en la eUICC.

1. **Funcion**

Las función del SM-DP+ permanece sin cambios, incluyendo:

- Almacenamiento seguro y cifrado de perfiles.
- Cifrado individualizado de cada perfil según el destino.
- Provisión autenticada y controlada de los perfiles.
- Gestión del estado de los perfiles (instalado, activado, desactivado, etc.).

La diferencia clave es que el destino final del perfil es ahora directamente la eUICC, donde el IPA embebido puede validar y activar el perfil sin intervención de componentes externos.

2. **Protocolos y seguridad**

Se mantiene el uso de:

- TLS 1.2/1.3.
- Infraestructura de Clave Pública (PKI).
- Comandos APDU encapsulados sobre HTTPS.
- Interfaces GSMA ES9+ y ES10b.

No obstante, al residir el IPA dentro de la eUICC, los mecanismos de autenticación y autorización se validan desde dentro del chip, ofreciendo una ejecución de operaciones más segura y ajustada al principio de confianza mínima.

3. **Flujo de comunicación**

El flujo general de provisión de perfiles también sigue el modelo original, con una diferencia esencial en el punto final de control:

1. El eIM solicita al SM-DP+ la descarga de un perfil en una eUICC específica.
2. El SM-DP+ cifra el perfil con las claves públicas del ISD-R/ISD-P embebidos en la eUICC.
3. El perfil es enviado directamente a la eUICC, donde el IPA embebido gestiona su instalación y activación.
4. El SM-DP+ recibe confirmación del éxito de la operación desde el eIM y, opcionalmente, desde la eUICC.

5. El backend eSIM registra el evento en blockchain y actualiza los estados asociados.

Este flujo es más directo y autónomo, y fortalece el modelo de zero-touch provisioning al concentrar la lógica de control dentro de la eUICC.

4.2.2.5. eIM (eSIM IoT Remote Manager)

La estructura interna del bloque eIM en esta arquitectura (ver figura 4.20) se conserva en gran medida respecto a la descrita en la sección [4.1.2.2]. Se mantienen componentes clave como el módulo de gestión de perfiles, los módulos de comunicación con IPA y SM-DP+, y los mecanismos de seguridad.

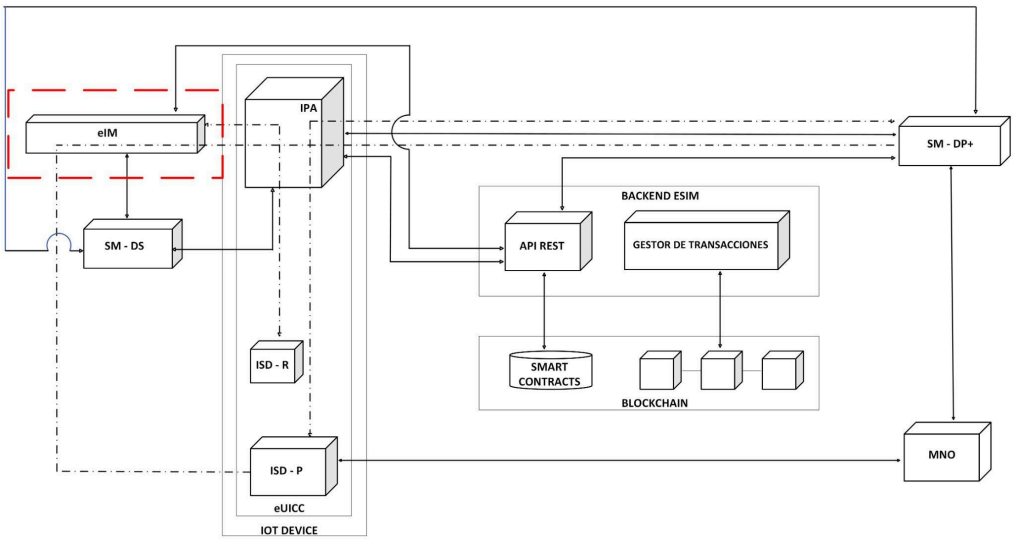

Figura 4.20. Bloque eIM.

Sin embargo, la diferencia central radica en que el IPA ya no reside en el dispositivo IoT, sino en la eUICC, lo que implica que todas las operaciones de gestión ahora son canalizadas directamente hacia el chip. Esta modificación refuerza la independencia de la eSIM respecto del sistema operativo del dispositivo, y permite una mayor estandarización y robustez en entornos heterogéneos.

1. **Funcion**

Las funciones del eIM siguen siendo esencialmente las mismas, gestión remota del estado de perfiles, coordinación de descargas y activaciones, y trazabilidad de operaciones. Sin embargo, en esta arquitectura:

- El eIM ya no necesita interactuar con el sistema operativo del dispositivo IoT, sino que se comunica directamente con la eUICC a través del canal seguro disponible (por ejemplo, HTTPS o CoAP, dependiendo del tipo de conectividad del dispositivo).
- Las respuestas del IPA ya no requieren atravesar capas intermedias del dispositivo, lo que reduce la latencia y posibles fallos por dependencias externas.
- Esta diferencia introduce una mejora significativa en robustez operativa y portabilidad entre dispositivos de diferentes fabricantes o sistemas operativos.

2. Protocolos

Se mantienen los mismos protocolos detallados previamente:

- CoAP/MQTT/TLS/DTLS para comunicaciones eficientes con dispositivos.
- HTTPS con el backend y el SM-DP+.
- Formatos de datos ASN.1 y JSON.

La única modificación relevante es que los comandos y respuestas viajan directamente entre el eIM y la eUICC, de manera agnóstica sin importar el sistema operativo o arquitectura de hardware del dispositivo IoT.

3. Mecanismos de Seguridad

Todos los mecanismos de seguridad descritos anteriormente se conservan. La principal mejora es que, al estar el IPA embebido en la eUICC:

- Las operaciones se ejecutan dentro de un entorno de ejecución seguro (Secure Element).
- Las claves criptográficas nunca abandonan la eUICC.
- Se minimiza la superficie de exposición frente a ataques que podrían aprovechar vulnerabilidades del sistema operativo de los dispositivos.

4. Flujo de Comunicación

El flujo operativo del eIM se mantiene, pero con los siguientes ajustes:

1. El eIM recibe una solicitud de gestión de perfil desde el OSS/BSS.
2. Consulta directamente el estado de la eUICC, sin depender del sistema operativo del dispositivo.
3. Se comunica con el backend eSIM para registrar la transacción, aplicar lógica de negocio y emitir el evento blockchain correspondiente.
4. Si es necesario, coordina con el SM-DP+ la entrega del perfil cifrado a la eUICC.
5. Ordena al IPA embebido (dentro de la eUICC) ejecutar la acción requerida (activación, eliminación, etc.).
6. Recibe respuesta directa desde el IPA/eUICC.
7. Actualiza el backend eSIM para cerrar el ciclo de trazabilidad y reflejar el resultado.

4.2.2.6. Mobile Network Operator (MNO)

El rol y funcionamiento general del MNO (ver figura 4.21) en esta arquitectura se mantienen prácticamente idénticos a los descritos en la sección [4.1.2.5] de la arquitectura anterior. Continúa siendo el proveedor de conectividad, responsable de generar los perfiles eSIM y autorizar su descarga a través del SM-DP+, en coordinación con el eIM y el backend eSIM.

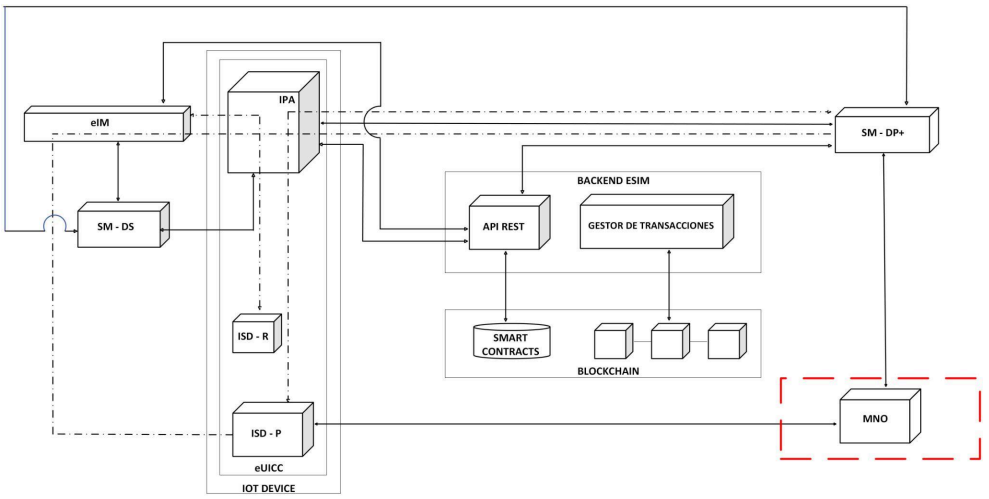

Figura 4.21. Bloque MNO.

Diferencias clave en esta arquitectura

Desacoplamiento del dispositivo IoT, al estar el IPA embebido directamente en la eUICC, el proceso de activación de perfil en el dispositivo IoT se vuelve más

directo, eliminando potenciales dependencias del sistema operativo del dispositivo. Esto permite al MNO ofrecer servicios a una gama más amplia de dispositivos con menor esfuerzo de compatibilidad.

Mayor trazabilidad a nivel de chip, dado que las operaciones son gestionadas dentro del entorno seguro de la eUICC, el MNO obtiene garantías adicionales sobre la integridad del proceso de descarga y activación del perfil. Esto mejora la confianza en el cumplimiento normativo y contractual del proceso de provisión.

Posibilidad de integraciones más finas vía backend eSIM, si bien las APIs entre MNO y backend se mantienen, el hecho de que las decisiones de activación puedan ahora depender directamente de eventos internos del chip (por ejemplo, respuestas del IPA dentro de la eUICC) permite al backend del operador enriquecer su lógica de negocio con datos más precisos y verificables.

4.2.2.7. SM-DS (Subscription Manager – Discovery Server)

El rol del SM-DS en esta arquitectura (ver figura 4.22) se mantiene sustancialmente igual al descrito en la sección [4.1.2.6] de la arquitectura anterior. Continúa siendo el componente encargado de intermediar entre el SM-DP+ y los dispositivos IoT, notificando la disponibilidad de nuevos perfiles listos para ser descargados.

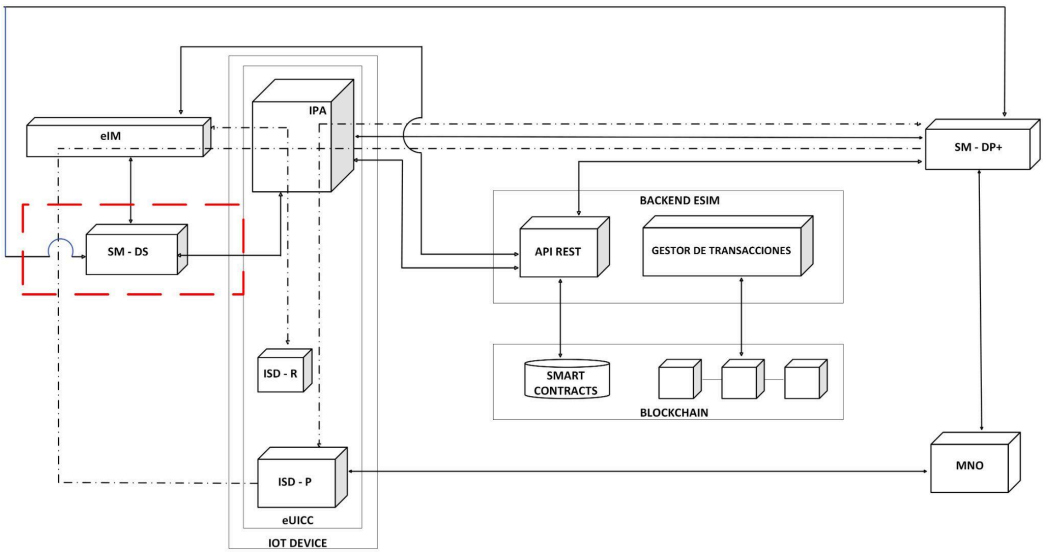

Figura 4.22. Bloque SM-DS.

Diferencias clave en esta arquitectura

- Destino del mensaje: eUICC con lógica embebida

A diferencia del escenario anterior donde el IPA estaba implementado sobre el dispositivo, en esta arquitectura el SM-DS ya no se comunica con el sistema operativo del dispositivo, sino directamente con la eUICC, que contiene internamente el módulo IPA. Esto cambia el punto de entrada del flujo de provisión y permite un proceso más autónomo, gestionado íntegramente dentro del chip.

- Mayor control dentro de la eUICC

Como la lógica de recepción, verificación de notificaciones y descarga del perfil se gestiona dentro del IPA embebido en la eUICC, se elimina la dependencia del dispositivo para consultar el SM-DS. Esto hace el proceso más seguro, estandarizado y desacoplado del sistema operativo, ideal para dispositivos IoT de bajo nivel o sin stack de red avanzado.

- Integración más estricta con eventos blockchain

Aunque el SM-DS en sí no interactúa directamente con la blockchain, las acciones que desencadena sí están sujetas a trazabilidad: cada consulta válida realizada por la eUICC puede originar una transacción registrada por el backend eSIM como parte del proceso de aprovisionamiento.

Desarrollados cada uno de los componentes que conforman esta arquitectura y después de haber comparado y obtenido diferencias con la arquitectura de blockchain e IPA en el dispositivo IoT, se muestra en la figura 4.23 un flujo de descarga de un perfil invocado desde el eIM.

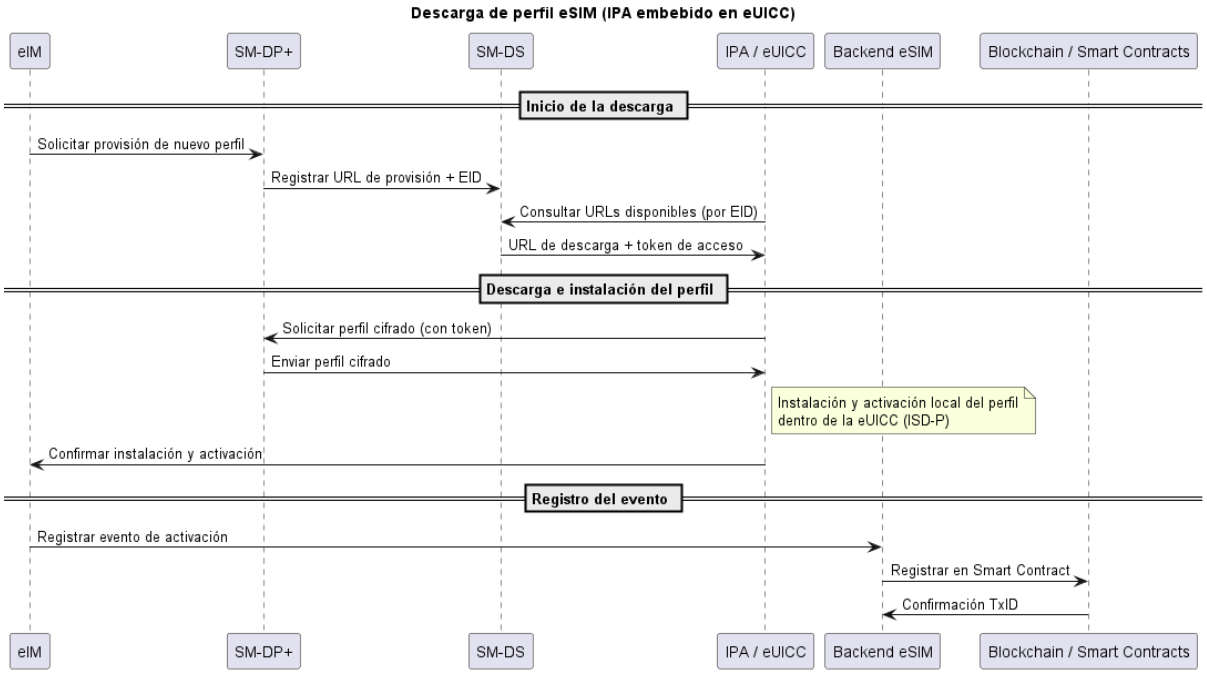

Figura 4.23. Descarga de perfil solicitado por eIM.

Este flujo representa la descarga de un perfil eSIM iniciado desde el eIM (eSIM IoT Remote Manager), que actúa como orquestador externo. Aunque la descarga es coordinada por el eIM, la gestión local del perfil se realiza íntegramente dentro de la eUICC, ya que el IPA (IoT Profile Assistant) está embebido en ella. El eIM solicita al SM-DP+ la provisión de un nuevo perfil, y este, a su vez, registra una URL segura en el SM-DS. El IPA, desde la eUICC, consulta esa URL utilizando su identificador EID y descarga el perfil cifrado directamente del SM-DP+. Una vez recibido, lo instala y activa localmente, notificando al eIM. Finalmente, la operación es registrada en la blockchain a través del backend eSIM para garantizar trazabilidad y transparencia.

Para complementar un poco más cómo funciona esta arquitectura, ahora se muestra un flujo de cambio de operador de red móvil, en un dispositivo que que ya tiene una eSIM operativa y registrada en la red, el cambio de operador puede generarse por distintos motivos de los cuales se pueden mencionar:

- Niveles de cobertura de red deficientes y acceso a tecnologías no disponibles por parte del proveedor actual
- Cambios en el modelo de negocio según costes y servicios del proveedor actual
- Acceso a punto de acceso de red (APN) con configuraciones particulares que el proveedor actual no puede proveer

- Acceso a plataformas de gestión con funcionalidades avanzadas.
- Mala experiencia de uso y nivel de soporte deficiente.

A continuación se muestra en la figura 4.24 el flujo de cambio de operador.

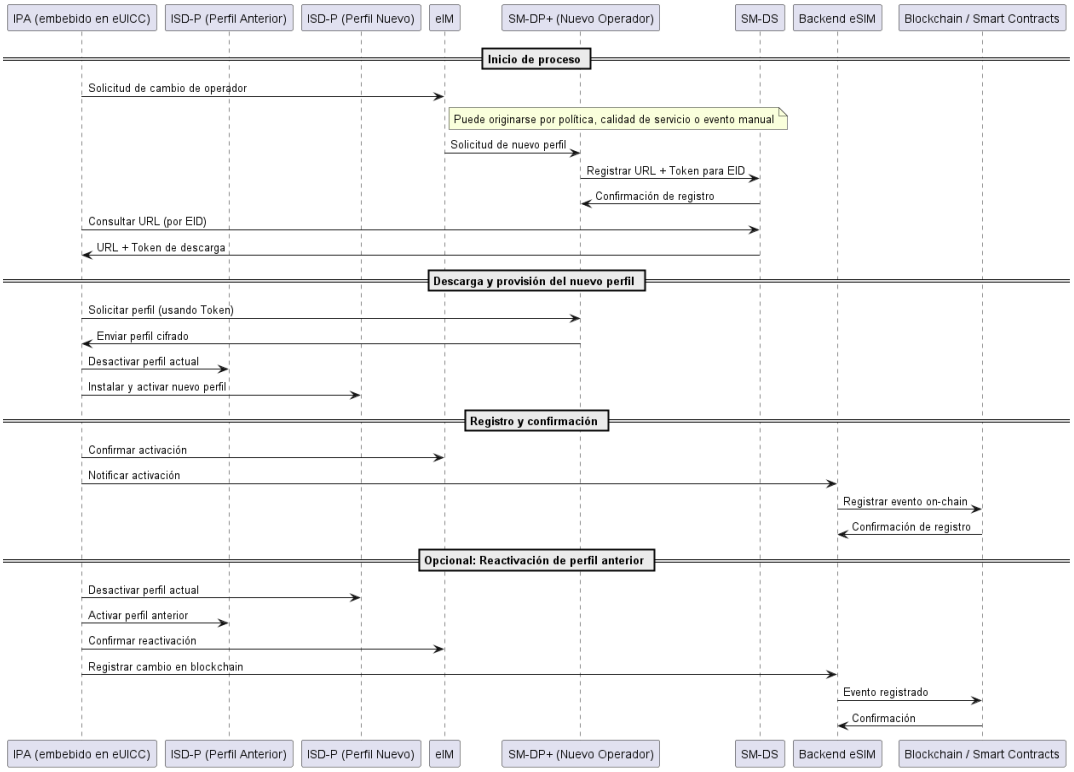

Figura 4.24. Flujo de cambio de operador.

El flujo de cambio de operador evidencia la versatilidad de esta arquitectura al permitir, mediante el eIM, la solicitud remota de perfiles alternativos cuando las condiciones de conectividad o negocio así lo requieren. El SM-DP+ del nuevo operador genera un perfil, lo cifra con la clave pública de la eUICC y lo pone a disposición del dispositivo a través del SM-DS, que actúa como punto de descubrimiento. El IPA, al estar embebido en la eUICC, consulta la URL y descarga de forma segura el nuevo perfil, desactivando previamente el perfil en uso para garantizar que solo uno permanezca activo. Este proceso no solo es compatible con escenarios de automatización, sino que también permite revertir el cambio y reactivar perfiles anteriores cuando sea necesario. La trazabilidad completa de la operación queda registrada en la blockchain, reforzando la transparencia del modelo.

En conjunto, esta arquitectura demuestra una evolución significativa frente al modelo tradicional, al integrar de forma nativa la lógica de gestión de perfiles en el

propio chip eUICC. La incorporación del IPA como componente embebido reduce latencias y dependencias externas, permitiendo decisiones locales rápidas y seguras. A su vez, la conexión directa con blockchain garantiza que cada operación quede registrada de forma inmutable, habilitando un esquema confiable de gobernanza y auditoría. Esta aproximación no solo optimiza los tiempos de provisión y cambio de operador, sino que también abre la puerta a escenarios de conectividad autónoma, resiliente y transparente para dispositivos IoT de nueva generación. En definitiva, representa un paso firme hacia una gestión más inteligente y descentralizada del ciclo de vida de las eSIM.

Capítulo 5 - Validación de la Propuesta

El presente capítulo tiene como objetivo validar la viabilidad y aplicabilidad de la arquitectura propuesta desde dos perspectivas complementarias. Por un lado, se presenta una prueba de concepto técnica, desarrollada en un entorno local controlado, que permitió simular y verificar los principales flujos operativos de la solución, tales como el registro de perfil y cambio de operador. Por otro lado, se presenta una evaluación cualitativa a través de una encuesta dirigida a profesionales con experiencia en tecnologías de telecomunicaciones, blockchain, ciberseguridad e IoT, quienes aportaron su visión sobre el valor, la factibilidad y el grado de innovación del enfoque propuesto. Ambos mecanismos de validación buscan aportar evidencia empírica y conceptual que respalda la propuesta, así como identificar posibles mejoras o áreas de aplicabilidad futura.

5.1. Prueba de concepto

El objetivo de esta prueba de concepto (PoC) fue validar la viabilidad técnica de los componentes innovadores de la arquitectura propuesta, en este caso el backend para orquestar operaciones del ciclo de vida de una eSIM y registrarlas de forma segura, auditable e inmutable en una red blockchain. El alcance de la prueba se acotó a la interacción entre el backend eSIM, los smart contracts desplegados en una blockchain local, y los clientes simulados que representan las peticiones de los actores del ecosistema propuesto como puede llegar a ser el IoT Profile Assistance (IPA). Ver diagrama de la POC en la figura 5.1

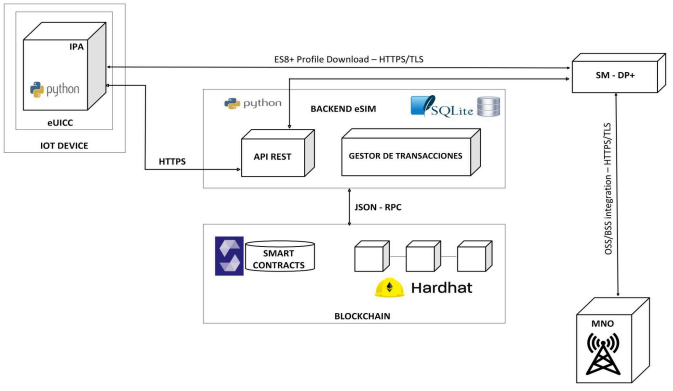

Figura 5.1. Diagrama de prueba de concepto.

5.1.1. Entorno y herramientas utilizadas

La prueba de concepto fue desarrollada localmente, utilizando herramientas que permitieron representar el flujo propuesto en las arquitecturas. A continuación, se detallan los principales componentes y herramientas utilizadas:

- **Backend eSIM**: Desarrollado en Python 3.12.3, utilizando el microframework Flask para la exposición de endpoints RESTful. Este backend orquesta el flujo de mensajes entre los distintos componentes y gestiona la persistencia de eventos.

- **Base de datos local**: Se utilizó SQLite3 como sistema de almacenamiento ligero y embebido para gestionar información sobre dispositivos, perfiles descargados y eventos registrados. Su simplicidad facilitó la implementación rápida de operaciones.

- **Blockchain local**: Se desplegó una red simulada utilizando Hardhat (v2.22.19), una herramienta de desarrollo para Ethereum, que permitió compilar, testear e interactuar con smart contracts sin depender de una testnet pública.

- **Smart Contracts**: los contratos inteligentes fueron escritos en **Solidity** (v0.8.29), implementando funciones de registro y trazabilidad de eventos asociados a la provisión de perfiles eSIM. Estos contratos son invocados desde el backend mediante Ethers.js.

- **Simulación de componentes**: Los roles del eIM y del IPA se simularon mediante el cliente curl para peticiones manuales y un script en Python (ipa_sim.py) para flujos más complejos que requerían la firma de transacciones.

5.1.2. Preparación del entorno

Primeramente instalamos todas las herramientas mencionadas anteriormente en nuestro entorno de trabajo WSL. Para visualizar los códigos utilizados dirigirse al siguiente repositorio de Github https://github.com/johmolin/TFM_MIOT_JM

Haciendo uso de hardhat en nuestro entorno local, creamos una blockchain la cual no está expuesta a una red pública, para ejecutarla lanzamos el siguiente comando.

```
npx hardhat node
```

Este comando lanza una blockchain en localhost, simulando una red de Ethereum para realizar pruebas sin necesidad de conectarse a redes públicas. Este nodo local genera cuentas con saldo simulado y permite desplegar contratos y ejecutar transacciones de forma instantánea. Expone un servidor JSON-RPC en http://127.0.0.1:8545, lo que facilita la interacción desde scripts o aplicaciones backend. Es ideal para entornos de desarrollo, ya que acelera la validación de flujos y lógica de los smart contracts. Además, muestra en consola todos los eventos y transacciones para facilitar la depuración, como se muestra a continuación en la figura 6.2:

```
johanmolina@DESKTOP-O3D3ON2:~/hardhat-moonbase$ npx hardhat node
Started HTTP and WebSocket JSON-RPC server at http://127.0.0.1:8545/

Accounts
========

WARNING: These accounts, and their private keys, are publicly known.
Any funds sent to them on Mainnet or any other live network WILL BE LOST.

Account #0: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266 (10000 ETH)
Private Key: 0xac0974bec39a17e36ba4a6b4d238ff944bacb478cbed5efcae784d7bf4f2ff80

Account #1: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8 (10000 ETH)
Private Key: 0x59c6995e998f97a5a0044966f0945389dc9e86dae88c7a8412f4603b6b78690d

Account #2: 0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC (10000 ETH)
Private Key: 0x5de4111afa1a4b94908f83103eb1f1706367c2e68ca870fc3fb9a804cdab365a

Account #3: 0x90F79bf6EB2c4f870365E785982E1f101E93b906 (10000 ETH)
Private Key: 0x7c852118294e51e653712a81e05800f419141751be58f605c371e15141b007a6

Account #4: 0x15d34AAf54267DB7D7c367839AAf71A00a2C6A65 (10000 ETH)
Private Key: 0x47e179ec197488593b187f80a00eb0da91f1b9d0b13f8733639f19c30a34926a

Account #5: 0x9965507D1a55bcC2695C58ba16FB37d819B0A4dc (10000 ETH)
Private Key: 0x8b3a350cf5c34c9194ca85829a2df0ec3153be0318b5e2d3348e872092edffba

Account #6: 0x976EA74026E726554dB657fA54763abd0C3a0aa9 (10000 ETH)
Private Key: 0x92db14e403b83dfe3df233f83dfa3a0d7096f21ca9b0d6d6b8d88b2b4ec1564e

Account #7: 0x14dC79964da2C08b23698B3D3cc7Ca32193d9955 (10000 ETH)
Private Key: 0x4bbbf85ce3377467afe5d46f804f221813b2bb87f24d81f60f1fcdbf7cbf4356

Account #8: 0x23618e81E3f5cdF7f54C3d65f7FBc0aBf5B21E8f (10000 ETH)
Private Key: 0xdbda1821b80551c9d65939329250298aa3472ba22feea921c0cf5d620ea67b97

Account #9: 0xa0Ee7A142d267C1f36714E4a8F75612F20a79720 (10000 ETH)
Private Key: 0x2a871d0798f97d79848a013d4936a73bf4cc922c825d33c1cf7073dff6d409c6
```

Figura 5.2. Despliegue de nodo blockchain local con Hardhat.

Desde el entorno local con Hardhat también nos permite compilar y desplegar los smart contracts con lenguaje de solidity, como se muestra a continuación en la figura 6.3:

```
johanmolina@DESKTOP-O3D3ON2:~/hardhat-moonbase$ npx hardhat compile
Compiled 1 Solidity file successfully (evm target: paris).
```

Figura 5.3. Compilación de smart contract con Hardhat.

Desplegando el smart contract en la blockchain como se muestra en la figura 5.4:

```
johanmolina@DESKTOP-O3D3ON2:~/hardhat-moonbase$ npx hardhat run scripts/deploy_identity.js --network localhost
Desplegando contratos con la cuenta: 0xf39Fd6e51aad88F6F4ce6aB8827279cfffFb92266
✅ ESIMRegistry desplegado en: 0x5FbDB2315678afecb367f032d93F642f64180aa3
```

Figura 5.4. Despliegue de smart contract localmente con Hardhat.

Una vez compilado y desplegado el contrato en la blockchain local, se confirma el éxito de la transacción por consola desde donde se está ejecutando el node de blockchain, ver figura 5.5:

```
eth_accounts
hardhat_metadata (20)
eth_accounts
hardhat_metadata (20)
eth_blockNumber
eth_getBlockByNumber
eth_feeHistory
eth_maxPriorityFeePerGas
eth_sendTransaction
  Contract deployment: ESIMRegistry
  Contract address:    0x5fbdb2315678afecb367f032d93f642f64180aa3
  Transaction:         0x9cf04d71095aa03de3563283400a8f9331c3e240cfe27eebcfe21b2a553cf67c
  From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
  Value:               0 ETH
  Gas used:            1280006 of 30000000
  Block #1:            0xd37502ac83feda0a8ad20ced0debfe8278b14f78773b3b8aedc1126843b13d74
```

Figura 5.5. Smart contract desplegado en la blockchain.

Para ver detalles del smart contract ESIMRegistry dirigirse al siguiente enlace de github https://github.com/johmolin/TFM_MIOT_JM/tree/main/contracts

Posteriormente se ejecuta el backend diseñado en Python haciendo uso del microframework flask, ver figura 5.6.

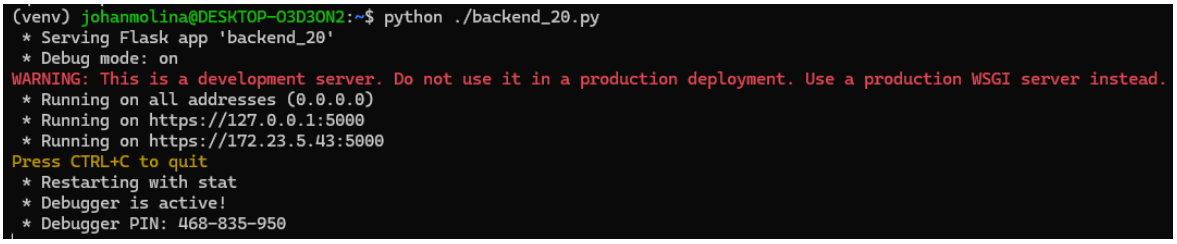

Figura 5.6. Ejecución del backend.

Al inicializar el backend, se crea la base de datos en SQLite3, que genera una tabla que guarda los registros de cada dispositivo, basado en eID, ICCID y operador.

Con el entorno preparado, se procede con la ejecución de operaciones principales como el registro de un dispositivo y perfil para posteriormente pasar a una segunda etapa con cambio de operador.

5.1.3 Registro de un dispositivo

Vamos a registrar los siguientes dispositivos:

eID (Identificador de eUICC)	ICCID (Integrated Circuit Card Identifier	Operador
89049032000000000001	8954312235915885l0	Vodafone
89049032000000000002	8954312235915885l1	Orange
89049032000000000003	8954312235915885l2	Telefónica
89049032000000000004	8954312235915885l3	Digi
89049032000000000005	8954312235915885l4	Claro
89049032000000000006	8954312235915885l5	Personal
89049032000000000007	8954312235915885l6	Personal
89049032000000000008	8954312235915885l7	Entel
89049032000000000009	8954312235915885l8	Telcel
89049032000000000010	8954312235915885l9	Movistar

Tabla 5.1. Dispositivos a registrar

Registro # 1

A través de comando curl:

```
johanmolina@DESKTOP-O3D3ON2:~$ curl -k -u admin:1234 -X POST https://127.0.0.1:5000/register_identity -H "Content-Type: application/json" -d '{
  "eid": "89049032000000000001",
  "iccid": "8954312235915885l0",
  "mno": "Vodafone",
  "pubkey": "0xf39Fd6e51aad88F6F4ce6aB8827279cfffb92266"
}'
{
  "status": "success",
  "tx_hash": "f035d9e86528b1c9ce0cae8aa81ee4718d0aeab9e35569a3cd567d4f227721c7"
}
```

Figura 5.7. Registro de dispositivo #1

Verificación de registro en backend:

```
(venv) johanmolina@DESKTOP-O3D3ON2:~$ python ./backend_20.py
 * Serving Flask app 'backend_20'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on https://127.0.0.1:5000
 * Running on https://172.23.5.43:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 468-835-950
127.0.0.1 - - [07/Jun/2025 22:20:22] "POST /register_identity HTTP/1.1" 201 -
```

Figura 5.8. Confirmación de registro #1 en backend.

Confirmación de Registro en la base de datos:

```
johanmolina@DESKTOP-O3D3ON2:~$ sqlite3 esim_data.db "SELECT * FROM devices;"
1|89049032000000000001|8954312235915885l0|Vodafone
```

Figura 5.9. Confirmación de registro #1 en base de datos.

Confirmación de transacción en la blockchain:

```
eth_getTransactionByHash
eth_getTransactionReceipt
eth_blockNumber
eth_getTransactionCount
eth_chainId
eth_sendRawTransaction
  Contract call:       ESIMRegistry#registerDevice
  Transaction:         0xf035d9e86528b1c9ce0cae8aa81ee4718d0aeab9e35569a3cd567d4f227721c7
  From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
  To:                  0x5fbdb2315678afecb367f032d93f642f64180aa3
  Value:               0 ETH
  Gas used:            120777 of 2000000
  Block #2:            0xba64b770628dca8718e8e722ccca17b8d157b45b24d213bd51753d616eabc291
```

Figura 5.10. Confirmación de registro #1 en blockchain.

Registro # 2

A través de comando curl:

Figura 5.11. Registro de dispositivo #2

Verificación de registro en backend:

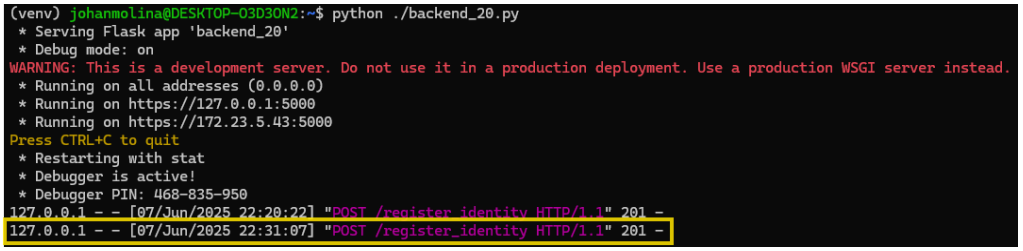

Figura 5.12. Confirmación de registro #2 en backend.

Confirmación de Registro en la base de datos:

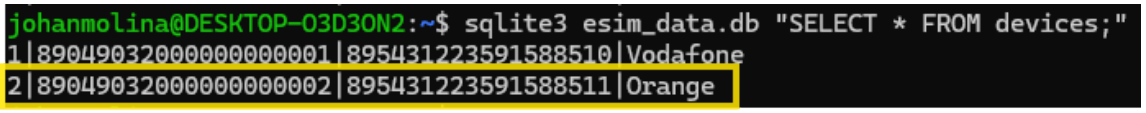

Figura 5.13. Confirmación de registro #2 en base de datos.

Confirmación de transacción en la blockchain:

Figura 5.14. Confirmación de registro #2 en blockchain.

Registro del resto de dispositivos de la tabla 5.1:

```
127.0.0.1 - - [07/Jun/2025 22:20:22] "POST /register_identity HTTP/1.1" 201 -
127.0.0.1 - - [07/Jun/2025 22:31:07] "POST /register_identity HTTP/1.1" 201 -
127.0.0.1 - - [07/Jun/2025 22:38:38] "POST /register_identity HTTP/1.1" 201 -
127.0.0.1 - - [07/Jun/2025 22:39:32] "POST /register_identity HTTP/1.1" 201 -
127.0.0.1 - - [07/Jun/2025 22:40:21] "POST /register_identity HTTP/1.1" 201 -
127.0.0.1 - - [07/Jun/2025 22:41:09] "POST /register_identity HTTP/1.1" 201 -
127.0.0.1 - - [07/Jun/2025 22:42:47] "POST /register_identity HTTP/1.1" 201 -
127.0.0.1 - - [07/Jun/2025 22:43:56] "POST /register_identity HTTP/1.1" 201 -
127.0.0.1 - - [07/Jun/2025 22:44:51] "POST /register_identity HTTP/1.1" 201 -
127.0.0.1 - - [07/Jun/2025 22:45:48] "POST /register_identity HTTP/1.1" 201 -
```

Figura 5.15. Verificación de todos los registros en backend.

```
johanmolina@DESKTOP-O3D3ON2:~$ sqlite3 esim_data.db "SELECT * FROM devices;"
1|89049032000000000001|895431223591588510|Vodafone
2|89049032000000000002|895431223591588511|Orange
3|89049032000000000003|895431223591588512|Telefonica
4|89049032000000000004|895431223591588513|Digi
5|89049032000000000005|895431223591588514|Claro
6|89049032000000000006|895431223591588515|Personal
7|89049032000000000007|895431223591588516|Personal
8|89049032000000000008|895431223591588517|Entel
9|89049032000000000009|895431223591588518|Telcel
10|89049032000000000010|895431223591588519|Movistar
```

Figura 5.16. Verificación de todos los registros en base de datos.

Antes de analizar los datos de consumo, es importante definir formalmente el concepto de gas en el contexto de Ethereum. Según la especificación técnica de la plataforma, el gas es la unidad fundamental que mide la cantidad de trabajo computacional requerido para ejecutar operaciones en la red. Su propósito principal es doble. Por un lado, sirve como mecanismo de control para evitar bucles infinitos o el abuso de recursos en una red; por otro, funciona como un incentivo económico para compensar a los validadores por los recursos (cómputo, almacenamiento) que aportan para procesar y securizar las transacciones.

Cada operación en la Máquina Virtual de Ethereum (EVM), desde una simple suma hasta el despliegue de un contrato inteligente, tiene un coste fijo en unidades de gas. Al enviar una transacción, el usuario debe especificar un límite de gas (gas limit), que es la cantidad máxima que está dispuesto a gastar, asegurando así que la ejecución no consuma recursos de forma indefinida [53]

A continuación, en la siguiente tabla 5.2 se muestra un recuento de consumo de gas en cada una de las operaciones realizadas:

eID (Identificador de eUICC)	ICCID (Integrated Circuit Card Identifier	Gas
89049032000000000001	895431223591588510	120777
89049032000000000002	895431223591588511	120753
89049032000000000003	895431223591588512	120801
89049032000000000004	895431223591588513	120729
89049032000000000005	895431223591588514	120741
89049032000000000005	895431223591588515	120777
89049032000000000007	895431223591588516	120777
89049032000000000008	895431223591588517	120741
89049032000000000009	895431223591588518	120753
89049032000000000010	895431223591588519	120777

Tabla 5.2. Consumo de gas por registro de dispositivo.

La Tabla 5.2 detalla el coste computacional, medido en unidades de gas, para cada una de las diez transacciones de registro de dispositivos ejecutadas en la red de pruebas. Como se puede observar, el consumo de gas para esta operación es notablemente estable, con una variación mínima entre las distintas transacciones. El coste se mantuvo en un rango muy estrecho, entre 120,729 y 120,801 unidades de gas.

El consumo medio de gas para el registro de un nuevo dispositivo en la blockchain fue de 120,762.5 unidades. Esta consistencia en el coste demuestra que la operación de registro definida en el contrato inteligente tiene un coste computacional predecible y eficiente, lo cual es un factor clave para evaluar la viabilidad económica y la escalabilidad de la solución en un entorno real.

Por otra parte, el sistema también detecta cuando se intenta registrar un eID que ya existe, con la finalidad de evitar duplicación de registros.

Figura 5.17. Detección de registro existente.

5.1.4. Cambio de operador

Para el cambio de operador, la acción es ejecutada desde el IPA, para ello se desarrolló un script en python para ver el código asociado dirigirse al siguiente enlace https://github.com/johmolin/TFM_MIOT_JM/tree/main/ipa

Para verificar este flujo se registraron los siguientes cambios de operador.

eID (Identificador de eUICC)	Nuevo ICCID	Nuevo Operador
89049032000000000001	895531223591588520	Orange
89049032000000000002	895531223591588521	Telefónica
89049032000000000003	895531223591588522	Claro
89049032000000000004	895531223591588523	Telcel
89049032000000000005	895531223591588524	Personal
89049032000000000006	895531223591588525	Entel
89049032000000000007	895531223591588526	Movistar
89049032000000000008	895531223591588527	Tigo
89049032000000000009	895531223591588528	Verizon
89049032000000000010	895531223591588529	Digi

Tabla 5.3. Cambios de operador.

Cambio de Operador #1

Ejecución del cambio de operador desde IPA

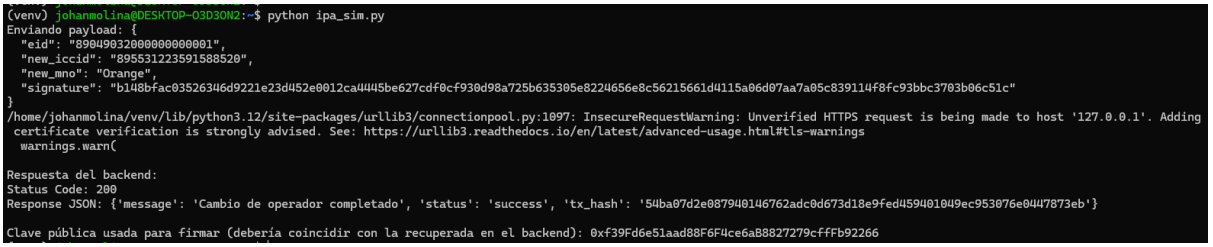

Figura 5.18. Cambio de operador desde IPA.

Confirmación de cambio de operador en backend:

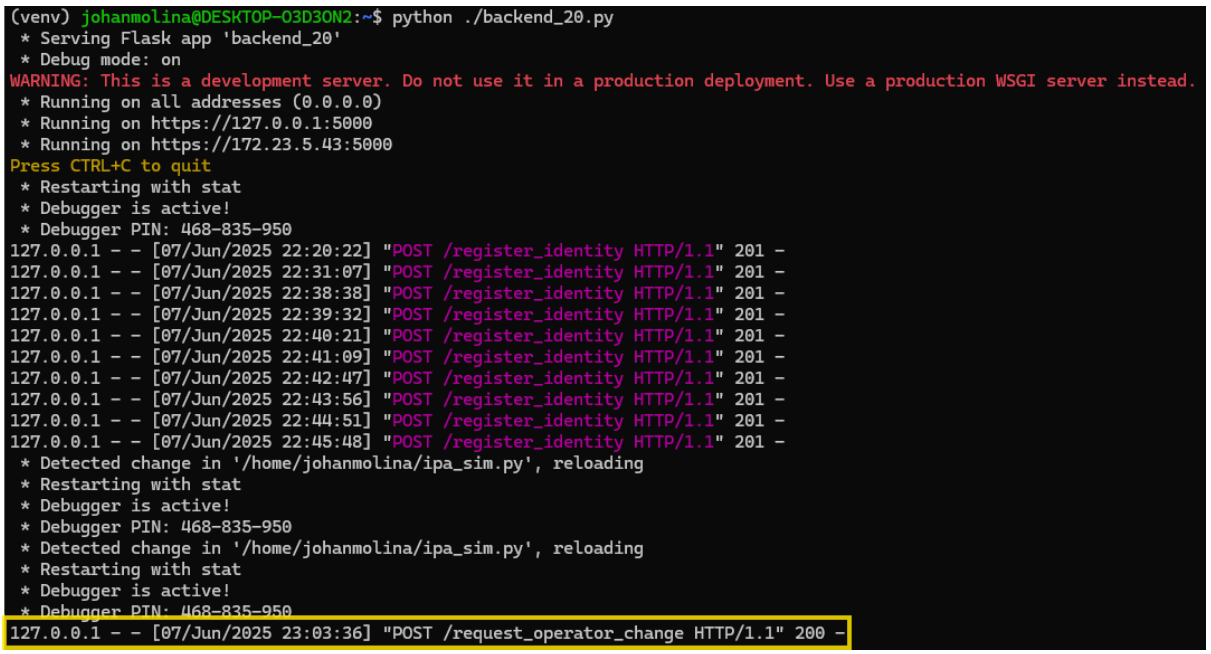

Figura 5.19. Registro de cambio de operador en backend.

Confirmación de cambio de operador en blockchain:

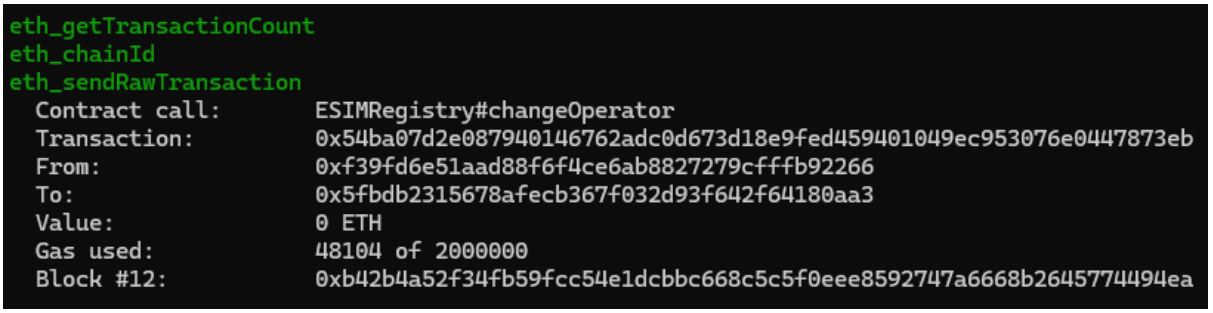

Figura 5.20. Registro de cambio de operador en blockchain.

Confirmación del cambio de operador en la base de datos:

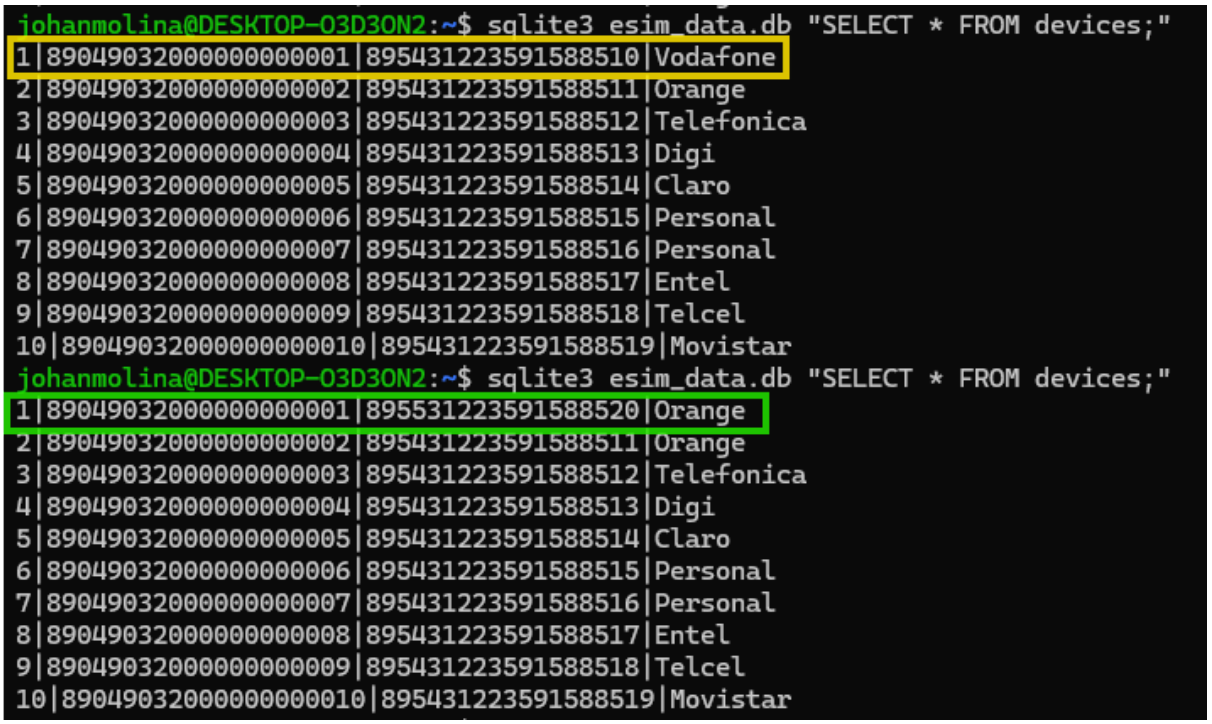

Figura 5.21. Registro de cambio de operador en base de datos.

Consulta de historial de cambios de operador por eID:

Figura 5.22. Consulta de historial de cambios por número de eID.

A continuación, en la siguiente tabla 5.4 se muestra el recuento de consumo de gas en cada una de las operaciones de cambio de operador:

eID (Identificador de eUICC)	Nuevo ICCID	Gas
89049032000000000001	895531223591588520	48104
89049032000000000002	895531223591588521	48152

89049032000000000003	895531223591588522	48092
89049032000000000004	895531223591588523	48104
89049032000000000005	895531223591588524	48128
89049032000000000006	895531223591588525	48092
89049032000000000007	895531223591588526	48128
89049032000000000008	895531223591588527	48080
89049032000000000009	895531223591588528	48116
89049032000000000010	895531223591588529	48080

Tabla 5.4. Consumo de gas por cambio de operador.

La Tabla 5.4 muestra el coste computacional en unidades de gas para la transacción de cambio de operador, una operación más compleja que el registro inicial, ya que implica la actualización de un estado existente en el contrato inteligente. Al igual que en el registro de dispositivos, los resultados demuestran una alta consistencia en el coste de la transacción, con valores que oscilan en un rango muy reducido, entre 48,080 y 48,152 unidades de gas.

El consumo medio para la operación de cambio de operador fue de 48,111.6 unidades de gas. Es interesante notar que, a pesar de implicar una lógica de actualización, el coste de esta operación es significativamente menor que el del registro inicial. Esto se debe a que la función changeOperator del contrato inteligente modifica un registro existente, lo cual es una operación computacionalmente más eficiente en la EVM que crear un registro completamente nuevo, que requiere más almacenamiento y, por tanto, más gas. Esta eficiencia y predictibilidad en el coste de las operaciones de actualización refuerzan la viabilidad del modelo para gestionar el ciclo de vida completo de los perfiles eSIM de manera sostenible.

Por otra parte, el sistema genera un error si se quiere realizar un cambio de operador que no cumple con el flujo establecido para este tipo de operación, ejemplo solo ejecuta de manera exitosa el cambio si pasa de un operador a otro, en caso de que se quiera realizar un cambio en el mismo operador se genera un error en la transacción.

Figura 5.23. Error en cambio de operador.

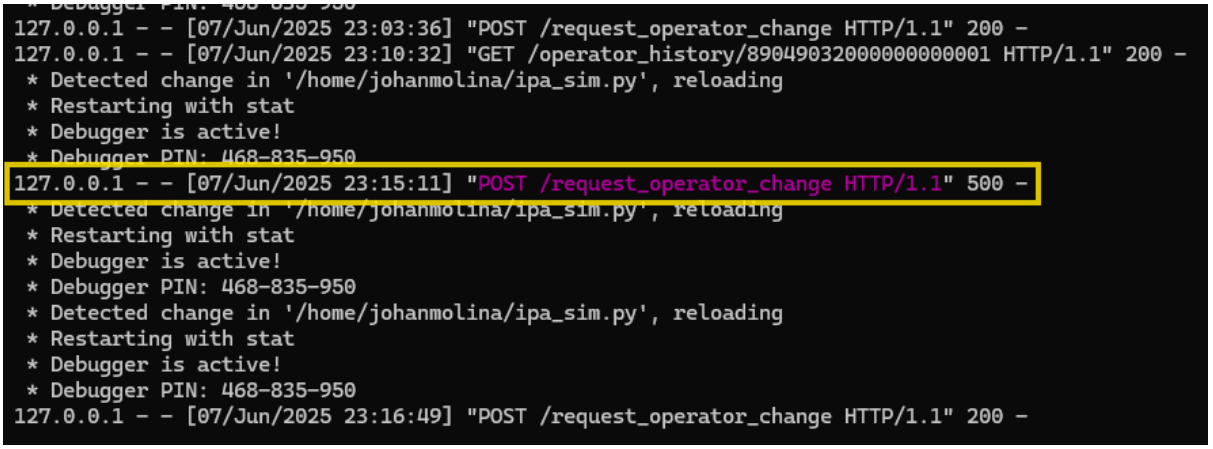

Figura 5.24. Error en cambio de operador en backend.

Figura 5.25. Error en cambio de operador en blockchain.

5.1.5. Casos de prueba ejecutados y resultados

Se validaron dos flujos críticos para demostrar la funcionalidad del sistema.

1. Caso de prueba: Registro de dispositivo e Identidad

- Ejecución: Se realizaron múltiples registros de dispositivos mediante peticiones POST con curl al endpoint /register_identity. Cada petición incluía el EID, ICCID, MNO y la clave pública del dispositivo.
- Resultados y Observaciones: La validación de esta prueba fue exitosa y se verificó en tres niveles:
    1. Backend: El servidor respondió con un código de estado 201 Created, confirmando la recepción y procesamiento correcto de la solicitud.
    2. Base de Datos: Una consulta directa a la base de datos esim_data.db demostró que los datos del nuevo dispositivo se persistieron correctamente en la tabla devices.
    3. Blockchain: Los logs del nodo de Hardhat confirmaron la ejecución exitosa del Contract call: ESIMRegistry#registerDevice, mostrando el hash de la transacción y el gas utilizado. Esto prueba el registro inmutable del evento.
- También se validó la lógica que impide registrar un EID duplicado, donde el sistema arrojó el error esperado.

2. Caso de prueba: Cambio de operador firmado por el IPA

- Ejecución: Se utilizó el script ipa_sim.py para simular la solicitud de un cambio de operador. El script construyó un mensaje con los datos del cambio, lo firmó con la clave privada del dispositivo (IPA) y lo envió al endpoint /request_operator_change.
- Resultados y Observaciones: Este flujo también se validó con éxito:
    1. Backend: El servidor verificó correctamente la firma digital, procesó la solicitud y respondió con un 200 OK y el hash de la nueva transacción.
    2. Base de Datos: Se confirmó que la tabla devices fue actualizada con el nuevo operador (Orange), y que se creó un nuevo registro en la tabla operator_changes para mantener el historial.
    3. Blockchain: Los logs de Hardhat mostraron la ejecución exitosa del Contract call: ESIMRegistry#changeOperator.
- Adicionalmente, se probó la lógica de negocio del smart contract al intentar cambiar a un dispositivo por el mismo operador que ya tenía, lo que provocó que la transacción en la blockchain fuera revertida correctamente con el mensaje de error: reverted with reason string 'Device already has this operator'.

5.1.6. Conclusiones de la prueba de concepto

La prueba de concepto realizada validó con éxito la hipótesis central de este trabajo: es técnicamente factible integrar una capa de blockchain para mejorar la trazabilidad y seguridad en la gestión de eSIMs. Se demostró que el backend desarrollado puede orquestar flujos, interactuar con una base de datos de estado y registrar eventos de forma inmutable en un smart contract, validando la lógica de negocio tanto en escenarios de éxito como de error controlado. Los datos de consumo de gas también aportan una primera métrica sobre el coste computacional de estas operaciones en una red Ethereum Virtual Machine (EVM).

5.1.7. Justificación del alcance de la prueba y consideraciones para un entorno real

1. Justificación del Alcance:

- **Foco en la innovación:** El principal aporte de este trabajo no es reinventar los componentes estándar del ecosistema eSIM (como SM-DP+, SM-DS), cuya funcionalidad ya está definida por la GSMA. La innovación reside en la introducción de una capa de confianza. Por tanto, la validación debe centrarse en demostrar que esta nueva pieza (el backend orquestador + blockchain) puede integrarse y aportar el valor prometido de trazabilidad,seguridad e interoperabilidad.
- **Viabilidad:** Re-implementar la infraestructura completa de un operador de telecomunicaciones es una tarea de enorme complejidad, fuera del alcance de un trabajo académico. Un enfoque pragmático y académicamente riguroso consiste en simular las interfaces de estos sistemas externos para probar la funcionalidad del núcleo de la propuesta, que es lo que se ha hecho en las pruebas.
- **Validación de la hipótesis principal:** Las pruebas ejecutadas, aunque acotadas, son suficientes para validar la hipótesis principal: que un evento como un cambio de operador puede ser iniciado por un componente, validado por un backend y registrado de forma inmutable en una blockchain, creando un sistema auditable y transparente.

2. Consideraciones para una implementación en un entorno real:

- **Seguridad avanzada:** En producción, se debería reemplazar la autenticación básica por protocolos más robustos. La clave privada del backend,

actualmente en un archivo .env, debería gestionarse a través de un HSM (Hardware Security Module) o un servicio de gestión de claves en la nube.

- **Elección de la red blockchain:** La PoC usa una red local. En un entorno real, se debería decidir entre:
    - Una **blockchain pública** (ej. Ethereum, Polygon, Polkadot), lo que implicaría costes de gas reales y consideraciones de privacidad de los datos on-chain.
    - Una **blockchain de consorcio o privada** (ej. Hyperledger Fabric), donde los actores (MNOs, fabricantes) actuarían como nodos de confianza, ofreciendo mayor privacidad y un coste transaccional menor.
- **Escalabilidad:** La base de datos SQLite debería ser reemplazada por un sistema de producción como PostgreSQL o una base de datos NoSQL, capaz de manejar un alto volumen de transacciones. El servidor Flask debería ejecutarse en un entorno robusto con balanceadores de carga y en alta disponibilidad.
- **Integración con sistemas reales:** La simulación de los componentes del ecosistema (SM-DP+, eIM) debería ser reemplazada por integraciones reales a través de las APIs estandarizadas por la GSMA.
- **Modelo de gobernanza:** Sería necesario definir un modelo de gobernanza claro para los smart contracts, quién puede desplegarlos, cómo se actualizan, quién paga el gas de las transacciones. Estas preguntas son críticas para la operatividad a largo plazo del sistema.

5.2 Evaluación con expertos

Para complementar la validación técnica realizada en la prueba de concepto, se llevó a cabo una evaluación cualitativa y cuantitativa mediante una encuesta dirigida a un grupo de profesionales del sector. El objetivo de esta evaluación era recabar la opinión experta sobre la relevancia del problema abordado, así como la viabilidad, utilidad e innovación de la solución propuesta, que integra la tecnología blockchain con los estándares de gestión de eSIM para IoT.

5.2.1. Metodología y perfil de los encuestados

Para complementar la validación técnica del prototipo, se llevó a cabo una evaluación cualitativa y cuantitativa mediante una encuesta a expertos. Se eligió

este método por su capacidad para recoger datos estructurados de una muestra amplia y geográficamente diversa de profesionales, permitiendo validar hipótesis específicas y, al mismo tiempo, capturar percepciones abiertas sobre la innovación y los desafíos de la propuesta.

Dada la naturaleza altamente especializada del campo de estudio, que requiere conocimientos convergentes en Telecomunicaciones, IoT, Ciberseguridad y Blockchain, se aplicó el método de muestreo por bola de nieve (snowball sampling)[73]. Esta técnica no probabilística resultó ideal para acceder a una red de profesionales de alto nivel, partiendo de un grupo inicial de contactos y ampliando la muestra a través de sus recomendaciones. Este enfoque permitió no solo alcanzar el tamaño de muestra deseado (57 profesionales), sino también asegurar la idoneidad y la alta especialización de los participantes.

El instrumento diseñado fue una encuesta online. Para garantizar que todos los participantes tuvieran el contexto adecuado antes de responder, el cuestionario comenzaba con una introducción que describe el problema abordado y el alcance de la solución propuesta, incluyendo además un enlace a un documento resumen con los detalles de la arquitectura. A continuación, la encuesta combinaba preguntas cerradas (utilizando una escala de Likert)[72] para obtener una valoración cuantitativa sobre la viabilidad y el valor de la solución, y preguntas abiertas, diseñadas para recoger opiniones cualitativas, identificar desafíos no contemplados y recibir sugerencias de mejora.

La encuesta fue respondida por un total de 57 profesionales. El perfil de los encuestados se caracteriza por una sólida experiencia en el sector:

Experiencia Profesional: El 55.4% de los participantes cuenta con más de 7 años de experiencia, un 23.2% tiene entre 4 y 7 años, y el 21.4% entre 1 y 3 años. Esta distribución garantiza que las valoraciones provienen mayoritariamente de perfiles senior con un profundo conocimiento del mercado y la tecnología.

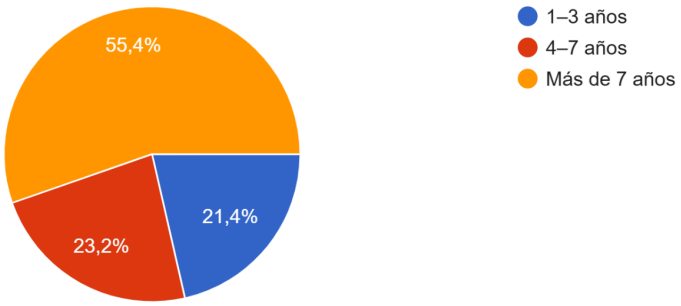

Figura 5.26. Años de experiencia de los encuestados.

Sector de Desempeño: Los participantes provienen de los sectores más relevantes para este trabajo, destacando Telecomunicaciones (75%), Ciberseguridad (10.7%) y Desarrollo de Software / IT (5.4%), entre otros.

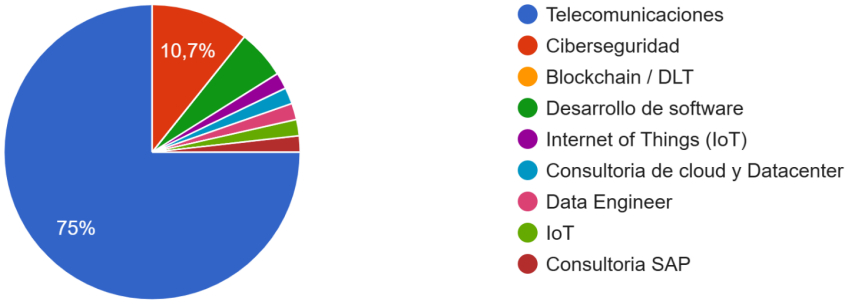

Figura 5.27. Sectores de desempeño de los encuestados.

Este perfil diverso y experimentado aporta solidez a los resultados obtenidos, validando la propuesta desde múltiples perspectivas (técnica, de negocio y de seguridad).

5.2.2. Resultados y análisis

Los resultados muestran una valoración generalmente favorable hacia la propuesta, destacando los siguientes puntos clave:

Trazabilidad y confianza: La afirmación "el uso de blockchain puede aportar valor en términos de trazabilidad y confianza" obtuvo una media de 4,26 sobre 5, con una baja dispersión, lo que evidencia un amplio consenso positivo.

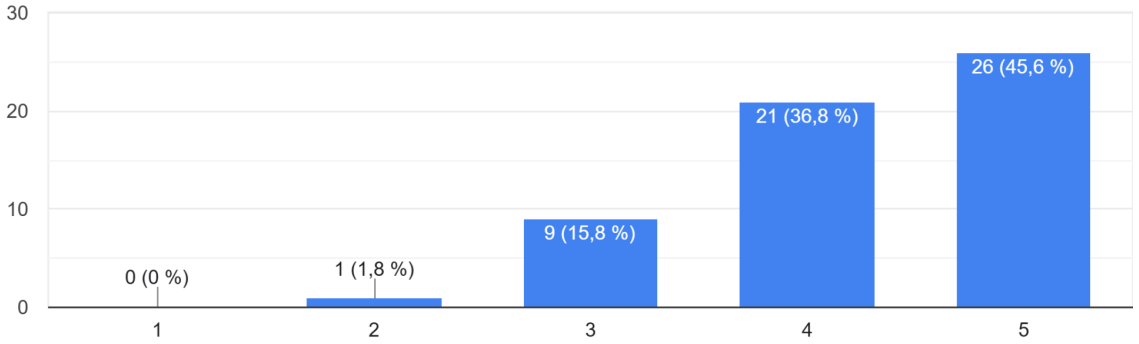

Figura 5.28. Nivel de aporte en trazabilidad y confianza.

Viabilidad técnica: La viabilidad de integrar blockchain en el ecosistema actual de eSIM fue valorada con una media de 3,89, lo que indica percepción moderadamente favorable, aunque acompañada de mayor variabilidad entre respuestas.

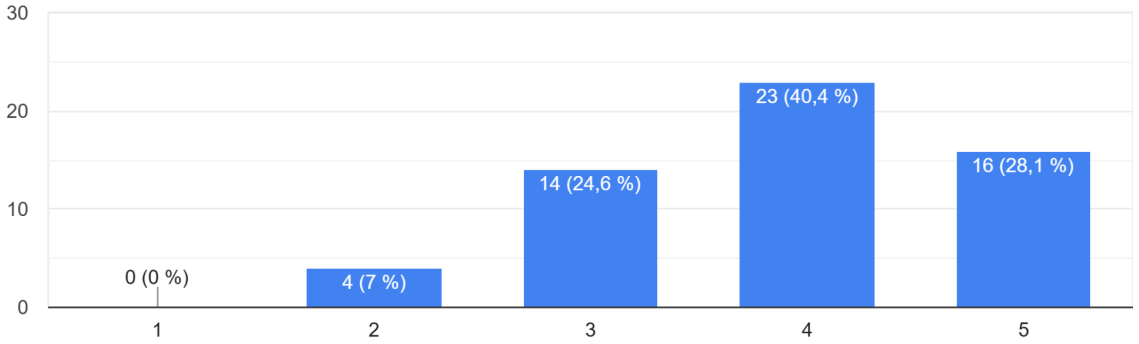

Figura 5.29. Viabilidad técnica.

Impacto en auditoría y operación: La mejora esperada en procesos como auditoría o resolución de disputas obtuvo una media de 4,14, reafirmando la utilidad operativa de la solución.

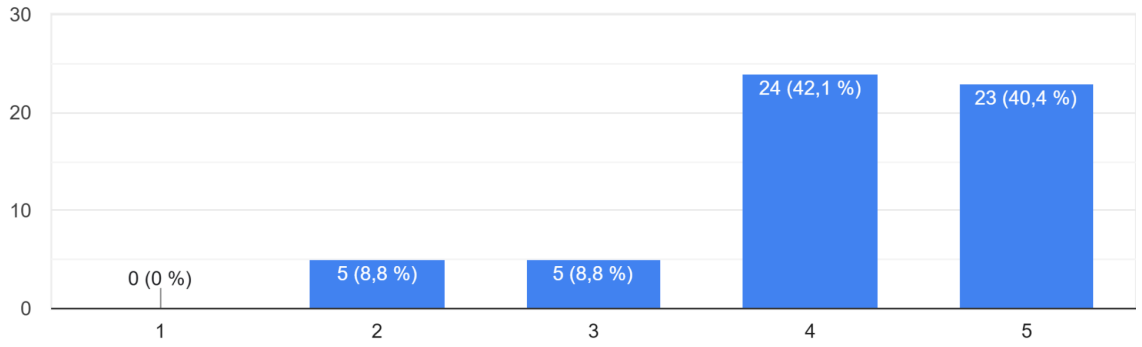

Figura 5.30. Impacto en auditoría y operación.

Recomendación organizacional: La pregunta sobre si se recomendaría explorar blockchain para la gestión de eSIMs alcanzó una media de 4,07, lo que refuerza la percepción de aplicabilidad más allá del entorno académico.

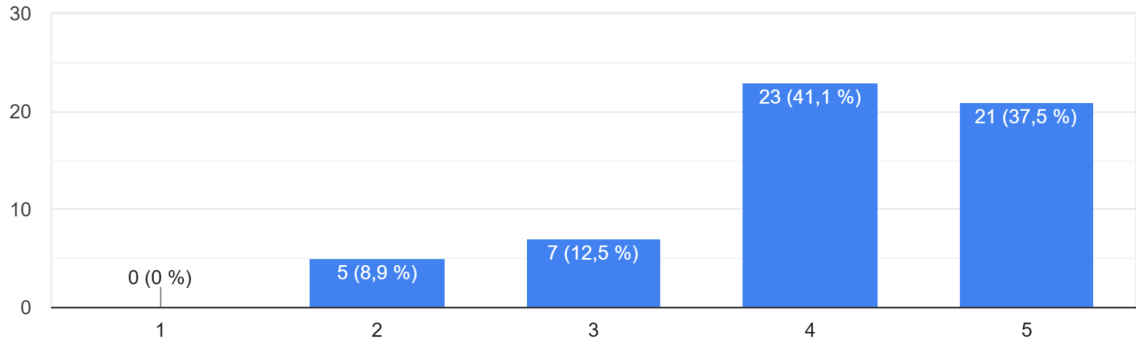

Figura 5.31. Recomendación organizacional.

En cuanto a las preguntas abiertas, se recogieron sugerencias que alertan sobre el consumo de recursos en dispositivos remotos, la necesidad de realizar pilotos controlados y la importancia de considerar aspectos legales y de privacidad.

Las respuestas de opción múltiple revelaron que las ventajas más reconocidas de una arquitectura distribuida son la transparencia, inmutabilidad de los registros y mejora en la auditoría.

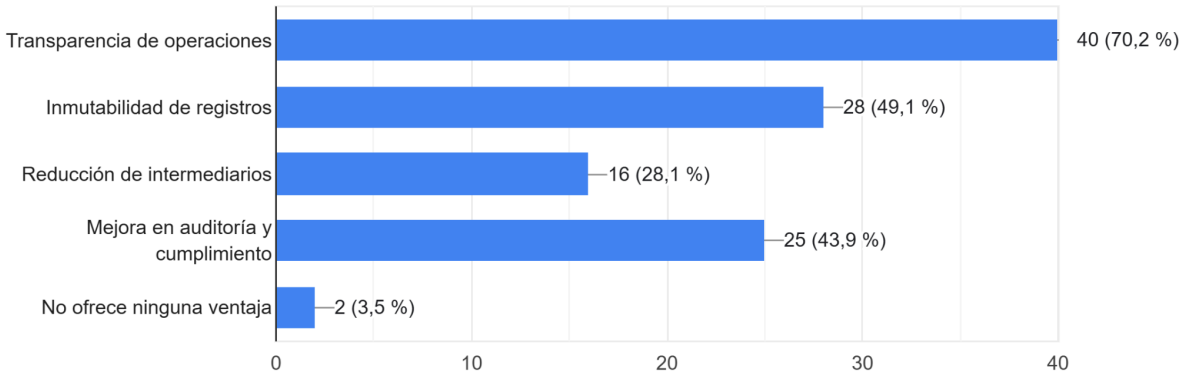

Figura 5.32. Ventajas de usar blockchain.

Por otra parte, los desafíos más señalados fueron la adopción, integración con sistemas actuales y el cumplimiento regulatorio.

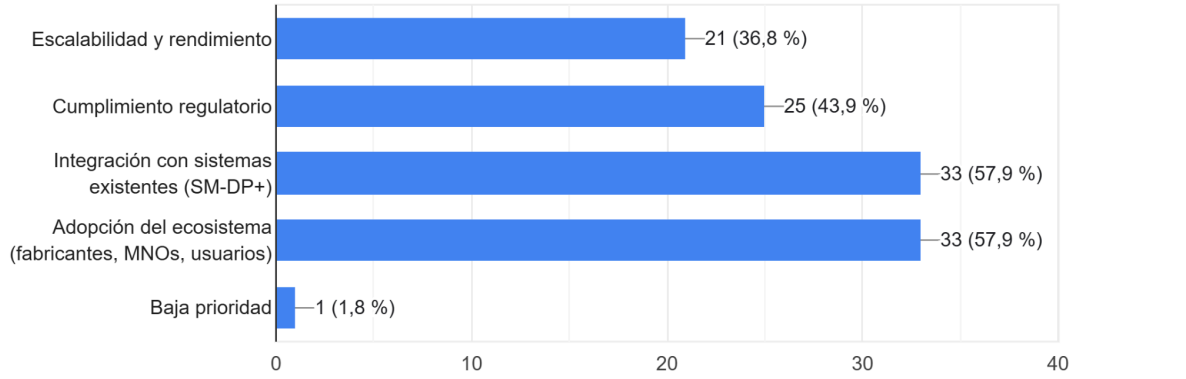

Figura 5.33. Desafíos al usar blockchain.

5.2.3. Conclusión de la validación cualitativa

La validación mediante encuesta revela que la comunidad experta reconoce un valor significativo en el uso de blockchain para la gestión de eSIMs, especialmente en términos de trazabilidad, transparencia y automatización. Aunque existen desafíos técnicos y regulatorios que deben ser abordados, los resultados confirman que la solución es percibida como viable, innovadora y aplicable a escenarios reales.

Capítulo 6 - Conclusiones y trabajo futuro

6.1. Conclusiones

Este trabajo fin de máster se inició con un análisis crítico de la problemática actual en torno a la provisión, seguridad e interoperabilidad de eSIMs en entornos IoT, donde se evidencian limitaciones en cuanto a trazabilidad, automatización y apertura tecnológica. A partir de esta motivación, se realizó una profunda investigación sobre estándares internacionales, modelos operativos actuales y las posibilidades que ofrece la tecnología blockchain como capa transversal de confianza. Este proceso permitió formular dos arquitecturas técnicas, respaldadas en estándares de la GSMA como SGP.31 y SGP.32, donde el componente IPA (IoT Profile Assistant) juega un rol central en la gestión segura de identidades y perfiles en un dispositivo IoT.

El desarrollo de una pequeña prueba de concepto, desplegada sobre una red blockchain local y un backend eSIM diseñado específicamente, permitió validar la viabilidad técnica de la propuesta. Paralelamente, se publicó una encuesta cualitativa/cuantitativa dirigida a profesionales con experiencia en telecomunicaciones, IoT y ciberseguridad, entre otros, quienes valoraron de forma positiva los principios técnicos, de seguridad y de escalabilidad que ofrece el modelo propuesto. Este enfoque práctico, combinado con el respaldo conceptual, aporta una propuesta que podría ser adaptable al entorno actual de los ecosistemas de IoT.

El principal aporte de este trabajo es la definición de una arquitectura de provisión y gestión de eSIMs para IoT que integra blockchain como tecnología de confianza, mejorando significativamente la trazabilidad, auditabilidad e interoperabilidad. Al eliminar la dependencia exclusiva de sistemas centralizados y facilitar una interoperabilidad más fluida entre actores, se ofrece un marco más transparente y automatizable para el aprovisionamiento de perfiles. Esta propuesta destaca por permitir el registro seguro de eventos críticos como cambios de operador, activaciones y autenticaciones de dispositivo, reduciendo el riesgo de fraude o pérdida de control sobre el ciclo de vida de la eSIM.

Entre las ventajas más relevantes se encuentran la posibilidad de automatizar reglas de cambio de operador basadas en lógica distribuida, la visibilidad completa de las transacciones por todos los actores involucrados, y la compatibilidad con entornos híbridos (centralizados y descentralizados). Además,

se habilita un modelo más resiliente, donde las decisiones sobre perfiles no dependen únicamente de un proveedor, sino que pueden ser verificadas y trazadas de forma pública o privada, según el caso. Esto supone una evolución importante respecto al modelo actual, alineándose con principios de descentralización, interoperabilidad y confianza que demanda el ecosistema IoT global hoy en día.

6.2. Trabajo futuro

Como proyección futura, resulta imprescindible profundizar en un análisis regulatorio detallado. Esto implica evaluar el encaje de esta propuesta dentro del marco normativo europeo (GDPR, eIDAS 2.0, NIS2), así como su compatibilidad con los requisitos técnicos definidos por la GSMA y otros organismos como ETSI o ENISA. También se abre la posibilidad de explorar modelos de gobernanza que definan roles claros para operadores, fabricantes, autoridades reguladoras y desarrolladores de infraestructura blockchain.

Otro eje de análisis será el coste de implementación de estas arquitecturas en redes comerciales, contemplando tanto el despliegue técnico (infraestructura, integración con sistemas OSS/BSS, soporte a dispositivos legacy), como el impacto económico sobre la cadena de valor. Asimismo, se plantea la necesidad de pruebas a mayor escala que permitan evaluar el rendimiento, la interoperabilidad entre distintas plataformas y la seguridad ante ataques avanzados. Finalmente, futuras líneas de investigación podrían enfocarse en extender este modelo a otros verticales industriales más allá de las telecomunicaciones, donde la gestión distribuida de identidades digitales y credenciales seguras es igualmente crucial.

Chapter 7 - Introduction

In today's digital age, connectivity and security of IoT (Internet of Things) devices are fundamental pillars for the development of intelligent and sustainable solutions. One of the most significant advancements in this area has been the evolution from the traditional SIM card, a physical card that enables device connectivity to mobile networks, to the eSIM (embedded SIM), a chip integrated directly into the device's hardware. This transition not only eliminates the need to insert or replace physical cards, but also enables the remote and secure downloading and management of multiple operator profiles. Thanks to this innovation, flexibility, efficiency, and scalability have been enhanced in environments with a high volume of connected devices. However, this evolution also introduces new challenges regarding data security, platform interoperability, and the reliable management of digital identities.

Blockchain technology, characterized by its decentralized nature and immutable, transparent history, could emerge as an innovative solution to address these challenges. Its application within the eSIM ecosystem for IoT devices strengthens the security of data transactions, ensures device authenticity, and provides a trustworthy mechanism for managing digital identities. Moreover, blockchain fosters interoperability among different providers and platforms, enabling a more secure and efficient environment without reliance on a central authority.

This study explores the application of blockchain technology to improve the security and interoperability of eSIMs in IoT devices. It analyzes key benefits, challenges, and use cases, with the objective of providing a conceptual framework that facilitates the adoption of these technologies in industrial and commercial settings. Through this analysis, the study aims to demonstrate how the combination of blockchain and eSIM can revolutionize connectivity and security in the rapidly expanding world of IoT.

7.1. Motivation

IoT environments are constantly expanding, with a growing number of connected devices across sectors such as smart cities, autonomous vehicles, asset management, telemetry, geolocation, agriculture, finance, and healthcare. However, scalability and interoperability of these solutions remain significant

challenges, particularly in terms of eSIM management and multi-operator connectivity.

Currently, there is no efficient and decentralized mechanism to manage eSIMs transparently across various mobile network operators without relying on centralized platforms. This fragmentation complicates eSIM profile management and hinders migration between providers, leading to inefficiencies and high operational costs. Additionally, the lack of traceability over device connection and operational history limits monitoring and auditing capabilities, affecting both network security and reliability.

Decentralizing eSIM management through blockchain would allow the creation of a distributed and auditable registry, removing dependency on operator-specific platforms and enhancing interoperability. A blockchain-based system would enable different operators to access a shared management environment, optimizing eSIM profile administration without relying on a single provider.

IoT device traceability and auditing is another aspect that can be improved through blockchain. Its ability to store immutable records would allow detailed tracking of connectivity history and operational events for each IoT device. This would not only enhance security, but also enable the detection of anomalies and potential fraud or misuse.

In terms of security and authentication, blockchain can provide robust mechanisms to reduce the risks of fraud or spoofing through verifiable digital identities and smart contracts. Each device could possess a unique and verifiable identity on the blockchain, ensuring that only authorized devices operate within the IoT network.

The purpose of this work is to evaluate the feasibility of these solutions and explore how blockchain can transform the way eSIMs and connectivity are managed in IoT devices, fostering a more secure, efficient, and transparent ecosystem.

It is important to emphasize that blockchain is not a universal solution. Its implementation in a production environment poses significant challenges in terms of operational costs, the definition of a clear governance model among stakeholders,

and its alignment with existing regulatory frameworks. Furthermore, the technology itself faces usability challenges [74], and its application is not always the most appropriate, requiring a well-justified use case [75].

In this context, the present research explores blockchain as a viable alternative to overcome the limitations of centralized models in eSIM management. While a traditional database controlled by a single actor perpetuates dependency and distrust among competitors, blockchain's core features—such as the ability to create an immutable and auditable ledger shared among all parties—offer an opportunity to design a more transparent and resilient system.

Therefore, this work does not propose blockchain as the only solution, but rather assesses its viability as a strategic tool to build a more open, interoperable, and trustworthy eSIM management ecosystem, which is the main objective of this research.

7.2. Objectives

The main objective of this work is to design a blockchain-based model that addresses the identified issues in eSIM management for IoT devices, optimizing security, interoperability, and traceability without relying on centralized platforms. To ensure that this solution is viable and aligned with industry standards, the integration with GSMA (Global System for Mobile Communications Association) standards will be explored, in order to assess how blockchain can complement or even replace certain functions to enhance efficiency and security.

The specific objectives of this work include:

- Developing a decentralized eSIM management model based on blockchain, enabling an IoT device to connect to multiple operators efficiently without relying on proprietary infrastructures.
- Implementing a traceability mechanism that immutably records the connection and operational history of each IoT device, allowing for more effective security audits and analyses.
- Designing a blockchain-based authentication system that guarantees each IoT device's unique and verifiable identity, reducing the risk of spoofing or unauthorized access.

- Exploring how blockchain can be integrated with GSMA standards (SGP.31, SGP.32) to improve eSIM security and interoperability in IoT environments, or alternatively, evaluating a model that replaces certain functions defined by these standards.

This approach aims to develop an innovative and feasible solution, grounded in solid technical foundations and ready for deployment in real-world environments. The proposal not only addresses current challenges but also provides tangible improvements in security and interoperability within the IoT ecosystem, ensuring compatibility with industry standards and global regulations.

7.3. Work Plan

The development of this work followed a structured plan divided into five key phases, executed over a six-month period:

Research and theoretical foundation phase: An extensive review of the theoretical background was carried out, including blockchain technology, the eSIM ecosystem for IoT, and GSMA standards (mainly SGP.31 and SGP.32). The goal of this phase was to consolidate the knowledge base required for the project.

Problem analysis and definition phase: The limitations of current centralized models for eSIM management were analyzed to identify key challenges in security, interoperability, and traceability. This stage defined the unique value proposition and scope of the proposed solution, focusing on identity management, device authentication, and decentralized profile provisioning.

Architecture design phase: Two technical architectures forming the core of this work were designed. The design included the definition of components (IPA, backend, eIM), interoperability flows between ecosystem actors (device, service provider, MNO), and the logic of smart contracts to manage events on the blockchain.

Development and technical validation phase: A proof of concept (PoC) was implemented to validate the technical feasibility of the architecture. The prototype focused on demonstrating the interaction between the backend, the database, and the smart contracts to orchestrate operations such as device registration and operator switching, as detailed in Chapter 5.

Evaluation and conclusions phase: Finally, the proposal was evaluated from two perspectives: the technical results of the PoC were analyzed, and a qualitative validation was conducted through a survey of industry experts. This allowed conclusions to be drawn regarding the model's feasibility and challenges, as well as the drafting of the final report.

7.4. Document Structure

This document consists of seven chapters that describe the application of blockchain technology to the security and interoperability of eSIMs in IoT devices. Below is a brief summary of the contents of each chapter:

Chapter 1: Introduction

Presents the context and motivation behind the research, emphasizing the importance of security and interoperability in the IoT ecosystem using eSIMs. The objectives of the work are described, along with the work plan that guided the study.

Chapter 2: Theoretical Foundations

This section presents the essential foundations for understanding the research, including the principles of blockchain technology, its types, characteristics, and the use of smart contracts. Key concepts of IoT and eSIM technology are also addressed, along with GSMA standards SGP.31 and SGP.32. The state of the art analyzes approaches that integrate blockchain, eSIM, and secure protocols to improve IoT device management and authentication. Solutions focused on provisioning automation, decentralized authentication, and credential protection such as the IoT SAFE standard are highlighted. These proposals serve as the technical basis for the solution proposed in this work.

Chapter 3: Methodologies and Technologies

This chapter outlines the methodological approach adopted for the development of the work, structured around three dimensions: documentary research, experimental development, and technical validation. It describes the sources used for regulatory and technical analysis, as well as the tools and programming languages used to build the prototype. It also explains the criteria for

conducting the expert survey, the sampling model, and the role of academic guidance. Finally, the set of technologies and platforms used in both the design and implementation of the proposed system is detailed.

Chapter 4: Proposed Technical Architecture

This chapter analyzes how blockchain technology can improve eSIM management in IoT devices. Existing models are examined, and two blockchain-based architectures are proposed that allow for secure and remote eSIM profile management, optimizing interoperability among mobile operators. Use cases, benefits, and challenges of the proposed solution are also presented.

Chapter 5: Proposal Validation

This chapter presents the validation of the proposed architectures through two complementary approaches. First, the technical proof of concept (PoC) is documented, demonstrating the feasibility of backend and blockchain interaction for orchestrating key operations such as device registration and operator switching. Second, the results of a survey conducted with industry professionals are presented and analyzed to assess the viability, value, and level of innovation of the solution, providing both empirical and expert-based validation of the research.

Chapter 6: Conclusions and Future Work

This section summarizes the key findings of the work, highlighting the improvements in security and efficiency offered by blockchain in eSIM management. Remaining challenges are identified, and future lines of research are proposed to continue developing innovative solutions in this field.

Chapter 8 - Conclusions and Future Work

8.1. Conclusions

This Master's Thesis began with a critical analysis of the current challenges surrounding the provisioning, security, and interoperability of eSIMs in IoT environments, where limitations in traceability, automation, and technological openness are clearly evident. Driven by this motivation, an in-depth investigation was conducted into international standards, current operational models, and the potential of blockchain technology as a transversal layer of trust. This research led to the formulation of two technical architectures, grounded in GSMA standards such as SGP.31 and SGP.32, where the IPA (IoT Profile Assistant) component plays a central role in the secure management of identities and profiles on an IoT device.

The development of a small proof of concept, deployed on a local blockchain network and a specifically designed eSIM backend, validated the technical feasibility of the proposal. In parallel, a qualitative/quantitative survey was distributed to professionals with experience in telecommunications, IoT, and cybersecurity, among others. Respondents positively assessed the technical, security, and scalability principles offered by the proposed model. This practical approach, combined with strong conceptual foundations, results in a proposal that could be adapted to the current landscape of IoT ecosystems.

The main contribution of this work is the definition of an eSIM provisioning and management architecture for IoT that integrates blockchain as a trusted technology, significantly improving traceability, auditability, and interoperability. By eliminating exclusive reliance on centralized systems and enabling more seamless interoperability among actors, the proposal provides a more transparent and automatable framework for profile provisioning. It stands out by enabling secure registration of critical events such as operator changes, activations, and device authentications—reducing the risk of fraud or loss of control over the eSIM lifecycle.

Among the most relevant advantages are the ability to automate operator-switching rules based on distributed logic, full visibility of transactions by all involved parties, and compatibility with hybrid (centralized and decentralized) environments. Furthermore, the model supports greater resilience, where profile management decisions are not solely dependent on a single provider but can be

publicly or privately verified and traced as needed. This marks a significant evolution from the current model, aligning with the principles of decentralization, interoperability, and trust demanded by today's global IoT ecosystem.

8.2. Future Work

Looking ahead, it is essential to deepen the regulatory analysis of the proposed model. This includes evaluating its alignment with the European legal framework (GDPR, eIDAS 2.0, NIS2), as well as its compatibility with the technical requirements defined by GSMA and other organizations such as ETSI or ENISA. There is also an opportunity to explore governance models that establish clear roles for operators, manufacturers, regulatory authorities, and blockchain infrastructure developers.

Another area of analysis involves the implementation cost of these architectures in commercial networks, considering both the technical deployment (infrastructure, integration with OSS/BSS systems, legacy device support) and the economic impact across the value chain. In addition, large-scale testing is necessary to assess performance, interoperability across different platforms, and resilience against advanced cyberattacks. Lastly, future lines of research could explore how this model can be extended to other industrial verticals beyond telecommunications, where distributed identity and secure credential management are equally critical.

Referencias

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [En línea]. Disponible: https://bitcoin.org/bitcoin.pdf

[2] X. Xu, I. Weber, and M. Staples, *Architecture for Blockchain Applications*. Springer, 2019. [En linea]. Disponible: https://doi.org/10.1007/978-3-319-99058-3

[3] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain Technology Overview," National Institute of Standards and Technology, 2019. [En linea]. Disponible: https://doi.org/10.6028/NIST.IR.8202

[4] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, 2017, pp. 557–564. [En linea]. Disponible: https://doi.org/10.1109/BigDataCongress.2017.85

[5] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A Survey on the Security of Blockchain Systems," *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2020. [En linea]. Disponible: https://doi.org/10.1016/j.future.2017.08.020

[6] C. Cachin and M. Vukolić, "Blockchain Consensus Protocols in the Wild," arXiv preprint, arXiv:1707.01873, 2017. [En linea]. Disponible: ttps://arxiv.org/abs/1707.01873

[7] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain Technology: Beyond Bitcoin," Applied Innovation Review, vol. 2, pp. 6–19, 2016.

[8] E. Androulaki et al., "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in Proc. EuroSys '18, ACM, 2018. [En linea]. Disponible: https://doi.org/10.1145/3190508.3190538

[9] M. Risius and K. Spohrer, "A Blockchain Research Framework: What We (Don't) Know, Where We Go from Here, and How We Will Get There," Business & Information Systems Engineering, vol. 59, no. 6, pp. 385–409, 2017. [En linea]. Disponible: https://doi.org/10.1007/s12599-017-0506-0

[10] M. Swan, Blockchain: Blueprint for a New Economy. O'Reilly Media, 2015.

[11] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "FHIRChain: Applying Blockchain to Securely and Scalably Share Clinical Data," Comput. Struct. Biotechnol. J., vol. 16, pp. 267–278, 2018. [En linea]. Disponible: https://doi.org/10.1016/j.csbj.2018.07.004

[12] S. Mohanty, Blockchain for Business. Packt Publishing, 2018.

[13] N. Szabo, "The Idea of Smart Contracts," 1997. [En linea]. Disponible: https://nakamotoinstitute.org/the-idea-of-smart-contracts

[14] V. Buterin, "A Next-Generation Smart Contract and Decentralized Application Platform," Ethereum Whitepaper, 2014. [En linea]. Disponible: https://ethereum.org/en/whitepaper

[15] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," IEEE Access, vol. 4, pp. 2292–2303, 2016. [En linea]. Disponible: https://doi.org/10.1109/ACCESS.2016.2566339

[16] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F. Y. Wang, "Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends," IEEE Trans. Syst., Man, Cybern.: Syst., vol. 49, no. 11, pp. 2266–2277, 2019. [En linea]. Disponible: https://doi.org/10.1109/TSMC.2019.2895123

[17] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts," in Proc. IEEE Symp. Security Privacy, 2016, pp. 839–858. [En linea]. Disponible: https://doi.org/10.1109/SP.2016.55

[18] N. Atzei, M. Bartoletti, and T. Cimoli, "A Survey of Attacks on Ethereum Smart Contracts (SoK)," in Proc. 6th Int. Conf. Principles of Security and Trust (POST), LNCS, vol. 10204, Springer, 2017. [En linea]. Disponible: https://doi.org/10.1007/978-3-662-54455-6_8

[19] K. Werbach and N. Cornell, "Contracts Ex Machina," Duke Law Journal, vol. 67, no. 2, pp. 313–382, 2017. [En linea]. Disponible: https://scholarship.law.duke.edu/dlj/vol67/iss2/2/

[20] J. Zhang, N. Xue, and X. Huang, "A Secure System For Pervasive Social Network-Based Healthcare," IEEE Access, vol. 7, pp. 68264–68274, 2019. [En linea]. Disponible: https://doi.org/10.1109/ACCESS.2019.2918331

[21] R. G. Brown, J. Carlyle, I. Grigg, and M. Hearn, "Corda: An Introduction," R3 CEV, 2016. [En linea]. Disponible: https://www.r3.com/wp-content/uploads/2017/06/corda-introductory-whitepaper-final.pdf

[22] GSMA, "SGP.21 eSIM Consumer Architecture Specification.", 2023. [En linea]. Disponible: https://www.gsma.com/solutions-and-impact/technologies/esim/esim-specification/

[23] GSMA, "SGP.31 eSIM IoT Architecture and Requirements.", 2024. [En linea]. Disponible:

https://www.gsma.com/solutions-and-impact/technologies/esim/wp-content/uploads/2024/04/SGP.31-v1.2.pdf

[24] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," Computer Networks, vol. 54, no. 15, pp. 2787–2805, 2010. [En linea]. Disponible: https://doi.org/10.1016/j.comnet.2010.05.010

[25] R. Roman, J. Zhou, and J. Lopez, "On the Features and Challenges of Security and Privacy in Distributed Internet of Things," Computer Networks, vol. 57, no. 10, pp. 2266–2279, 2013. [En linea]. Disponible: https://doi.org/10.1016/j.comnet.2012.12.012

[26] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 2013. [En linea]. Disponible: https://doi.org/10.1016/j.future.2013.01.010

[27] X. Xu et al., "Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends," IEEE Trans. Syst., Man, Cybern.: Syst., vol. 49, no. 11, pp. 2266–2277, 2019. [En linea]. Disponible: https://doi.org/10.1109/TSMC.2019.2895123

[28] GSMA, "Remote SIM Provisioning (RSP) Architecture for consumer Devices", 2023. [En linea]. Disponible: https://www.gsma.com/solutions-and-impact/technologies/esim/esim-specification/

[29] GSMA, "SGP.32 eSIM IoT Additional Guidelines.", 2024. [En linea]. Disponible: https://www.gsma.com/solutions-and-impact/technologies/esim/gsma_resources/sgp-32-v1-2/

[30] GSMA, "Using the SIM as a 'Root of Trust' to Secure IoT Applications.", 2019. [En linea]. Disponible: https://www.gsma.com/solutions-and-impact/technologies/internet-of-things/wp-content/uploads/2019/12/IoT.04-v1-Common-Implementation-Guide.pdf

[31] P. Krishnan, K. Jain, S. R. Poojara, S. N. Srirama, T. Pandey, and R. Buyya, "eSIM and blockchain integrated secure zero-touch provisioning for autonomous cellular-IoTs in 5G networks," Computer Communications, vol. 216, pp. 324–345, 2024. [En linea]. Disponible: https://doi.org/10.1016/j.comcom.2023.12.023

[32] B. K. Mohanta, A. Sahoo, S. Patel, S. S. Panda, D. Jena and D. Gountia, "DecAuth: Decentralized Authentication Scheme for IoT Device Using Ethereum Blockchain," 2019. [En linea]. Disponible: https://ieeexplore.ieee.org/document/8929720

[33] T. Neudecker and H. Hartenstein, "Network layer aspects of permissionless blockchains," IEEE Commun. Surv. Tutor., vol. 21, no. 1, pp. 838–857, 2018.

[34] J. Chauhan, S. Seneviratne, Y. Hu, A. Misra, A. Seneviratne, and Y. Lee, "Breathing-based authentication on resource-constrained IoT devices using recurrent neural networks," Computer, vol. 51, no. 5, pp. 60–67, 2018.

[35] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for IoT devices," IEEE Internet Things J., vol. 6, no. 1, pp. 580–589, 2019.

[36] M. C. Chow and M. Ma, "A Secure Blockchain-Based Authentication and Key Agreement Scheme for 3GPP 5G Networks," Sensors, vol. 22, p. 4525, 2022. [En linea]. Disponible: https://doi.org/10.3390/s22124525

[37] 3GPP, "TS 33.501 – Security Architecture for 5G System.", 2018. [En linea]. Disponible: https://www.3gpp.org/DynaReport/33501.htm

[38] A. S. Ahmed, M. Thakur, S. Paavolainen, and T. Aura, "Transparency of SIM profiles for the consumer remote SIM provisioning protocol," Annals of Telecommunications, vol. 76, pp. 187–202, 2021. [En linea]. Disponible: https://doi.org/10.1007/s12243-020-00791-2

[39] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, IETF, 2018. [En linea]. Disponible: https://datatracker.ietf.org/doc/html/rfc8446

[40] ISO/IEC 7816-4:2020, "Integrated circuit cards – Organization, security and commands for interchange.", 2020. [En linea]. Disponible: https://www.iso.org/obp/ui/en/#iso:std:iso-iec:7816:-4:ed-4:v1:en

[41] NIST, "NIST SP 800-57 Part 1 Rev. 5 – Recommendation for Key Management", 2020 [En linea]. Disponible: https://csrc.nist.gov/publications/

[42] M. B. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, IETF, 2015. [En linea]. Disponible: https://datatracker.ietf.org/doc/html/rfc7519

[43] OASIS, "MQTT Version 5.0. OASIS Standard," 2019. [En linea]. Disponible: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

[44] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," RFC 7252, IETF, 2014. [En linea]. Disponible: https://datatracker.ietf.org/doc/html/rfc7252

[45] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," RFC 2104, IETF, 1997. [En linea]. Disponible: https://datatracker.ietf.org/doc/html/rfc2104

[46] JSON-RPC Working Group, "JSON-RPC 2.0 Specification.", 2013. [En linea]. Disponible: https://www.jsonrpc.org/specification

[47] Ethereum Foundation, "Web3.js Documentation.",2023. [En linea]. Disponible: https://web3js.readthedocs.io

[48] R. Ricmoo, "Ethers.js: A Complete Ethereum Wallet Implementation and Utilities in JavaScript.",2023. [En linea]. Disponible: https://docs.ethers.org

[49] Ethereum Foundation, "Solidity Documentation." [En linea]. Disponible: https://docs.soliditylang.org

[50] IPFS Project, "InterPlanetary File System (IPFS)." [En linea]. Disponible: https://docs.ipfs.tech

[51] NIST, "FIPS PUB 186-4: Digital Signature Standard (DSS).", 2013. [En linea]. Disponible: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

[52] NIST, "FIPS PUB 180-4: Secure Hash Standard (SHS).", 2015. [En linea]. Disponible: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

[53] G. Wood "Ethereum: A Secure Decentralised Generalised Transaction Ledger," Shanghai Version efc5f9a, 2025 [En linea]. Disponible: https://ethereum.github.io/yellowpaper/paper.pdf

[54] J. Molina Medina y S. Hassan Collado, "Resultados de Encuesta técnica – Evaluación de propuesta para Trabajo Final de Master (TFM). Tecnología Blockchain Aplicada a la Securización e Interoperabilidad de eSIMs en Dispositivos IoT". Zenodo, jun. 14, 2025. doi: 10.5281/zenodo.15664323

[55] 3GPP, "About 3GPP." [En línea]. Disponible: https://www.3gpp.org/about-us/introducing-3gpp

[56] Hassan, S., Brekke, J.K., Atzori, M., Bodó, B. "Scanning the European Ecosystem of Distributed Ledger Technologies for Social and Public Good", 2020 [En linea]. Disponible: https://publications.jrc.ec.europa.eu/repository/handle/JRC121675

[57] Filippi, P. D., & Hassan, S. "Blockchain technology as a regulatory technology: From code is law to law is code", 2016. [En linea]. Disponible: https://doi.org/10.5210/fm.v21i12.7113

[58] Hassan, S., & De Filippi, P. "Decentralized Autonomous Organization. Internet Policy Review", 2021. [En linea]. Disponible: https://doi.org/10.14763/2021.2.1556

[59] NIST, "SP 800-186 – Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters", 2023. [En linea]. Disponible: https://csrc.nist.gov/pubs/sp/800/186/final

[60] A. Koutsos, "The 5G-AKA Authentication Protocol Privacy," 2019.[En linea]. Disponible: https://ieeexplore.ieee.org/document/8806761

[61] Python Software Foundation, "Python Language Reference", 2024. [En línea]. Disponible: https://www.python.org
[62] Pallets, "Flask Documentation", 2024. [En línea]. Disponible: https://flask.palletsprojects.com
[63] D. R. Hipp, "About SQLite", SQLite Consortium, 2024. [En línea]. Disponible: https://www.sqlite.org/about.html
[64] Microsoft, "Windows Subsystem for Linux Documentation", 2024. [En línea]. Disponible: https://docs.microsoft.com/en-us/windows/wsl/
[65] Nomic Foundation, "Hardhat: Ethereum development environment for professionals", 2024. [En línea]. Disponible: https://hardhat.org/
[66] Remix Project, "Remix - Ethereum IDE", 2024. [En línea]. Disponible: https://remix-project.org/
[67] Web3.py, "Web3.py Documentation", 2024. [En línea]. Disponible: https://web3py.readthedocs.io/
[68] Object Management Group, "OMG Unified Modeling Language (UML)", 2017. [En línea]. Disponible: https://www.omg.org/spec/UML/
[69] OpenAI, "ChatGPT", 2024. [En línea]. Disponible: https://chat.openai.com/
[70] Google, "Gemini", 2024. [En línea]. Disponible: https://gemini.google.com/
[71] Google, "Google Docs", Google Workspace, 2024. [En línea]. Disponible: https://www.google.com/docs/about/
[72] R. Likert, "A Technique for the Measurement of Attitudes," Archives of Psychology, vol. 22, no. 140, pp. 1–55, 1932.
[73] L. A. Goodman, "Snowball Sampling," The Annals of Mathematical Statistics, vol. 32, no. 1, pp. 148–170, 1961.
[74] Jorge Saldivar, Elena Martínez-Vicente, David Rozas, Maria-Cruz Valiente, and Samer Hassan. "Blockchain (not) for Everyone: Design Challenges of Blockchain-based Applications". 2023. [En línea]. Disponible. https://doi.org/10.1145/3544549.3585825
[75] Tenorio-Fornés, Á., Hassan, S., & Pavón, J. "Peer-to-Peer System Design Trade-Offs: A Framework Exploring the Balance between Blockchain and IPFS. Applied Sciences", 2021. [En línea]. Disponible. https://doi.org/10.3390/app112110012
[76] M. Nieles, K. Dempsey, V. Yan Pillitteri "An Introduction to Information Security," NIST Special Publication 800-12r1, National Institute of Standards and Technology, Gaithersburg, MD, USA, Jun. 2017. [En línea]. Disponible: https://doi.org/10.6028/NIST.SP.800-12r1
[77] MetaMask, "A crypto wallet & gateway to blockchain apps," 2025. [En línea]. Disponible: https://metamask.io/

[78] Truffle Suite, "Truffle: The Ultimate Smart Contract Development Toolkit," 2025. [En línea]. Disponible: https://trufflesuite.com/

[79] E. Barker, L. Chen, A. Roginsky, A. Vassilev, and R. Davis, "Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography," NIST Special Publication 800-56A, Rev. 3, National Institute of Standards and Technology, Gaithersburg, MD, USA, Apr. 2018. [En línea]. Disponible: https://doi.org/10.6028/NIST.SP.800-56Ar3

[80] M. Burrows, M. Abadi, and R. Needham, "A logic of Authentication," ACM Transactions on Computer Systems (TOCS), vol. 8, no. 1, pp. 18–36, Feb. 1990. [En línea]. Disponible: https://doi.org/10.1145/77648.77649

[81] C. J. F. Cremers, "The Scyther Tool," 2025. [En línea]. Disponible: https://people.cispa.io/cas.cremers/scyther/

[82] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "The COOJA Simulator for the Contiki Operating System," in Proc. of the IPSN'06 Workshop on From Theory to Practice in Distributed Systems, Nashville, TN, USA, Apr. 2006.

[83] ConsenSys, "ConsenSys Quorum: The Enterprise-Grade Ethereum," 2025. [En línea]. Disponible: https://consensys.io/quorum