IOTPAY

Specification and Use Cases

DUE DATE	05/03/2025
SUBMISSION DATE	25/02/2025
TEAM	Werenode
VERSION	0.9
AUTHORS	François Chiron, Benoît Maïsseu, Evgenii Zhdarkin

This project has received funding from the European Union's Horizon Europe Research and Innovation programme under Grant Agreement No. 101061548

Experiment information

Team Name	IOTPAY
Coach	Juan Juan
EU Organization	Werenode
CA/US Organization	Gapask

EU EU Organization Members

- Benoit Maisseu
- Francois Chiron
- Evgenii Zhdarkin
- Lucas Goncalves
- Nadiya Khokhryakova

CAUS CA/US Organization Members

- Rajesh Murthy

Experiment description

IOTPAY is a decentralized, autonomous payment system designed for IoT devices, offering secure and feeless transactions. It uses Web3.0 technologies like smart contracts and embedded wallets to enhance security, automates M2M payments, and allows IoT devices to perform transactions autonomously, revolutionizing industries like smart grids, logistics, and autonomous vehicle payments.

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

Table of contents

List of Figures

ABBREVIATIONS

AI	Artificial Intelligence
DFD	Data Flow Diagram
DLT	Distributed Ledger Technology
DSO	Distribution System Operators
EV	Electric Vehicle
EVSE	Electric Vehicle Supply Equipment
IoT	Internet of Things
IP	Internet Protocol
MVP	Minimum Viable Product
OCPI	Open Charge Point Interface
OCPP	Open Charge Point Protocol
POC	Proof of Concept
TCP	Transmission Control Protocol

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

1 INTRODUCTION

In the rapidly evolving landscape of the Internet of Things (IoT), the need for secure, efficient, and autonomous payment solutions has become increasingly paramount. Traditional payment systems often involve intermediaries, leading to increased costs and potential delays. To address these challenges, Werenode is developing IOTPAY, an innovative payment solution designed specifically for IoT devices, leveraging blockchain technology to facilitate direct, autonomous transactions, taking into account the specific constraints and needs of IoT devices.

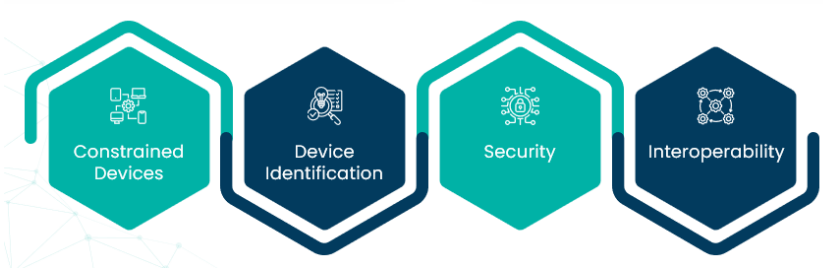

Figure 1 – IOT priority topics

IOTPAY operates by integrating digital wallets directly into IoT devices, enabling them to make and receive payments with minimal transaction costs and high speed. This approach eliminates the need for intermediaries, thereby reducing costs and enhancing transaction efficiency. The use of blockchain technology ensures that each transaction is secure, transparent, and immutable, fostering trust among all parties involved.

The applications of IOTPAY are vast and varied. In the realm of energy management for smart buildings, IOTPAY enables secure, value-driven energy transactions between IoT devices within a decentralized energy community, optimizing energy consumption and promoting sustainability. In logistics, it streamlines operations by automating payments for processes involving fees or digital documentation, thereby enhancing efficiency and reducing manual intervention. For connected cars, IOTPAY facilitates autonomous payments for services such as tolls, parking, and charging, enhancing the convenience and efficiency of transportation systems.

By providing a robust, blockchain-based payment infrastructure, IOTPAY addresses the unique challenges posed by the proliferation of IoT devices. Its decentralized nature not only reduces operational costs but also improves traceability and scalability, making it a comprehensive solution for the future of IoT payments. As the IoT ecosystem continues to expand, solutions like IOTPAY will play a crucial role in facilitating seamless and secure transactions between devices, paving the way for even more integrated and autonomous systems.

This document will further detail some use cases of the solution and its technical specification.

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

2 SYSTEM OVERALL CONCEPT

IOTPAY is an autonomous payment solution designed for Internet of Things (IoT) devices, built upon blockchain technology. Its system architecture comprises several key functional components:

- **Embedded Wallets**: Each IoT device is equipped with a digital wallet, enabling it to store, send, and receive payments directly. This integration allows devices to participate in transactions without relying on external payment gateways.
- **Smart Contracts**: IOTPAY utilizes blockchain-based smart contracts to automate and enforce transaction terms between devices. These contracts ensure that payments are executed only when predefined conditions are met, enhancing security and reducing the need for manual oversight.
- **Decentralized Public Key Infrastructure (DPKI)**: To maintain a secure and trustworthy environment, IOTPAY implements a DPKI tailored for IoT devices. This infrastructure manages device identities and authentication, ensuring that only authorized devices can engage in transactions.
- **Feeless Transactions**: By optimizing the blockchain protocol, IOTPAY facilitates transactions without traditional fees, making microtransactions economically viable and promoting seamless interactions among devices.

This cohesive system enables IoT devices to conduct secure, autonomous transactions, fostering a more efficient and interconnected IoT ecosystem.

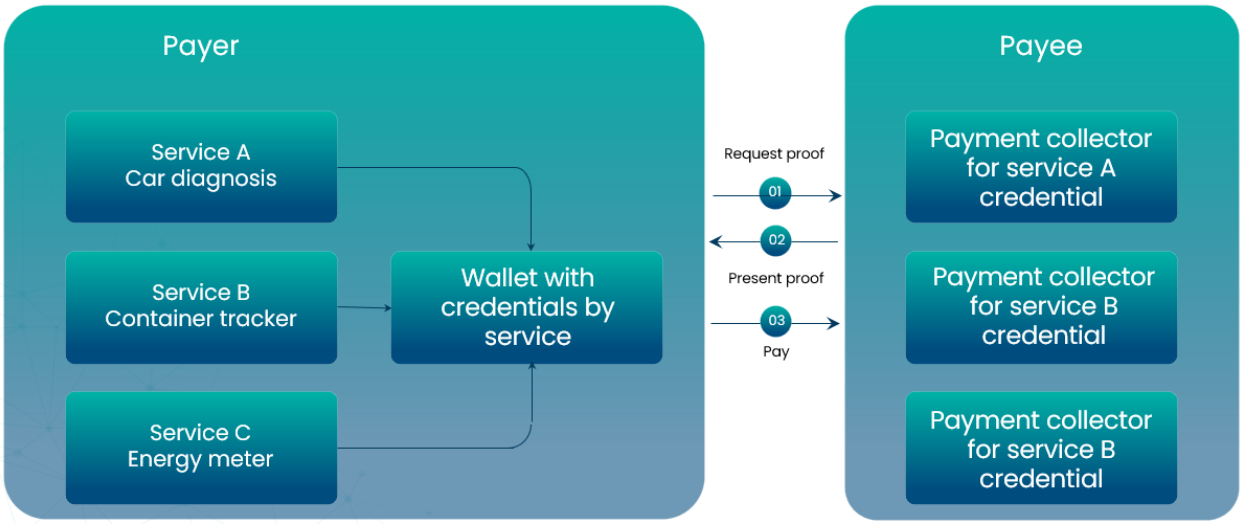

Figure 2 – System overall concept

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

IOTPAY is also composed of several integral software components that work together to facilitate decentralized, autonomous payments for IoT devices:

- Smart Contracts: These are blockchain-based agreements that automate and enforce the terms of transactions between IoT devices. They handle autonomous payments, issue and revoke certificates, and manage feeless transactions, ensuring secure and efficient interactions without the need for intermediaries.
- Embedded Wallets: Each IoT device is equipped with an embedded wallet capable of storing, sending, and receiving payments, as well as managing Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs). This integration allows devices to participate directly in the IOTPAY ecosystem, facilitating seamless and autonomous financial interactions.
- Web Application User Interface: A user-friendly web interface enables users to interact with the IOTPAY system, providing functionalities such as monitoring transactions, 8managing device credentials, and configuring payment settings. This interface ensures that users have comprehensive control and oversight of their IoT devices' payment activities.
- Mobile Application: Complementing the web interface, the mobile application offers on-the-go access to the IOTPAY system. Users can manage payments, receive notifications, and oversee device interactions directly from their mobile devices, enhancing convenience and accessibility.

These components collectively establish a robust framework for secure, efficient, and autonomous financial transactions within the IoT ecosystem, leveraging blockchain technology to eliminate intermediaries and reduce operational costs.

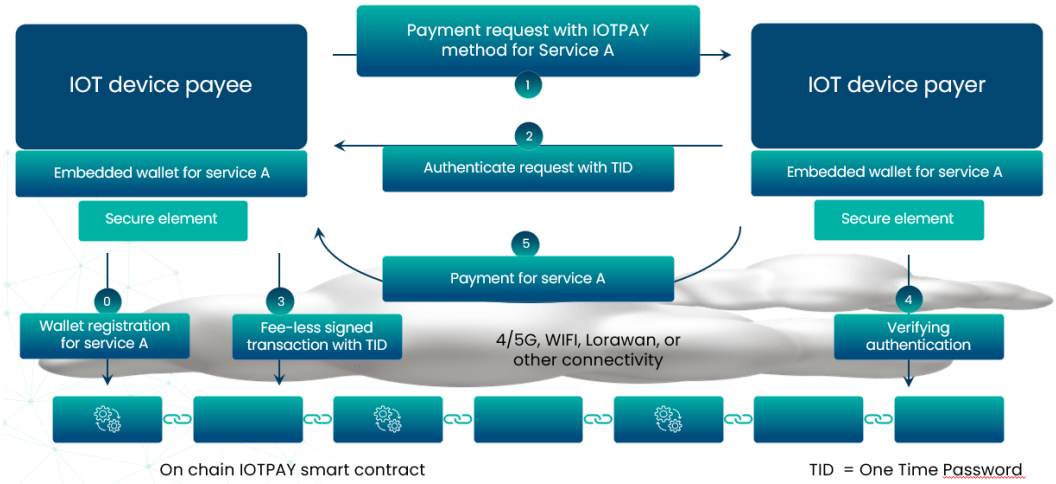

Figure 3 – IOTPAY protocol

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

3 USE CASES

3.1 Smart Buildings

In a smart building operating within a decentralized energy community, IOTPAY facilitates seamless and secure energy transactions among interconnected IoT devices. By leveraging blockchain technology, IOTPAY enables devices such as solar panels, energy storage systems, and smart appliances to autonomously trade energy based on real-time demand and supply conditions.

For instance, consider a scenario where a building's rooftop solar panels generate surplus energy. Through IOTPAY, this excess energy can be automatically sold to neighboring buildings or stored for future use. Each transaction is securely recorded on the blockchain, ensuring transparency and traceability. This decentralized approach not only optimizes energy consumption but also reduces reliance on centralized power grids, promoting sustainability and resilience within the community.

Moreover, IOTPAY's integration with IoT devices allows for real-time monitoring and management of energy resources. Smart meters and sensors continuously collect data on energy production and consumption, enabling predictive analytics and informed decision-making. This data-driven strategy enhances operational efficiency, reduces costs, and contributes to the overall well-being of the building's occupants.

By facilitating autonomous, secure, and efficient energy transactions, IOTPAY plays a pivotal role in advancing smart buildings and decentralized energy communities. Its implementation leads to optimized energy usage, cost savings, and a reduced environmental footprint, aligning with the goals of modern sustainable development.

3.2 Logistics

In the logistics sector, IOTPAY can significantly enhance operational efficiency by automating payment processes between Internet of Things (IoT) devices. Consider a scenario involving a fleet of delivery trucks equipped with IoT sensors for real-time tracking and monitoring. Traditionally, processing payments for services such as tolls, fuel, and maintenance require manual intervention, leading to potential delays and increased administrative costs.

With IOTPAY, each vehicle is integrated with an embedded wallet, enabling it to autonomously handle payments. For instance, as a truck passes through a toll booth, the embedded wallet

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

communicates directly with the toll system's IoT device, executing an immediate, feeless transaction. This process is facilitated by blockchain-based smart contracts, ensuring secure and transparent transactions without the need for intermediaries.

Additionally, IOTPAY's decentralized public key infrastructure (DPKI) manages device identities and authentication, ensuring that only authorized devices participate in transactions. This security measure is crucial in logistics, where safeguarding data integrity and preventing unauthorized access are paramount.

By automating these payment processes, IOTPAY reduces administrative overhead, minimizes human error, and accelerates transaction times. This leads to a more streamlined supply chain, allowing logistics companies to focus on core operations and improve service delivery.

3.3 Battery Swapping

In the context of electric vehicle (EV) battery swapping, IOTPAY can streamline and secure the payment process, enhancing the efficiency and user experience of battery-swapping services.

An EV driver arrives at a battery-swapping station to replace a depleted battery with a fully charged one. Traditionally, this process might involve manual payment methods, leading to potential delays and inconvenience.

The EV is equipped with the embedded IOTPAY digital wallet that communicates directly with the swapping station's system. This wallet facilitates automatic, feeless transactions, allowing for a seamless payment experience without manual intervention. Utilizing blockchain-based smart contracts, IOTPAY ensures that the payment is securely processed only when the battery swap is successfully completed. This automation reduces the risk of errors and enhances trust between the service provider and the user. IOTPAY's DPKI manages the authentication of both the vehicle and the swapping station, ensuring that only authorized entities participate in the transaction. This security measure protects against unauthorized access and potential fraud.

The automated payment process reduces the time spent at the swapping station, aligning with the quick turnaround times that battery swapping aims to achieve. For instance, companies like Nio have demonstrated battery swaps in approximately three minutes, and integrating IOTPAY can further streamline this process. Drivers can enjoy a hassle-free experience without the need to handle physical payments or wait for transaction approvals. Blockchain technology ensures that all transactions are transparent and immutable, providing a secure record of each battery swap.

By integrating IOTPAY into EV battery-swapping services, operators can offer a more efficient, secure, and user-friendly experience, encouraging the adoption of battery-swapping models and supporting the broader use of electric vehicles.

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

3.4 Smart Industry

A smart industrial machine wants to dynamically select the most cost-effective IoT service provider using the IOTPAY solution, so that it can optimize its operating costs in real time.

A smart industrial machine, such as a robotic arm in a factory, operates in an environment where multiple IoT service providers offer various services like predictive maintenance and energy monitoring at varying costs. The machine is registered within the IOTPAY system, equipped with a unique Decentralized Identifier (DID). Similarly, the IoT service providers are also registered and connected through Payment Credentials (PCs), which define the terms for transactions between the machine and each provider.

When the machine identifies a need for an IoT service, such as maintenance or energy monitoring, it dynamically begins the process of discovering available service providers. Using the IOTPAY system, the machine queries the providers, evaluating their offerings based on factors such as service quality, cost, and response time. Once the optimal provider is selected, the machine initiates a payment request, which formalizes the terms for accessing the selected service.

Next, a transaction query is dynamically configured to establish payment terms, including frequency and amount. The machine then consumes the chosen IoT service without any manual intervention, receiving the required data, energy or maintenance seamlessly (as examples).

Upon successful completion of the service, the IOTPAY system automatically triggers a feeless transaction using the embedded wallet within the machine. The smart contract verifies the transaction based on predefined criteria, such as whether the service was delivered as agreed. If the service quality declines or a better provider becomes available, the machine can dynamically switch providers by creating new transaction requests.

By leveraging the IOTPAY system, the machine effectively optimizes its operational costs by always selecting the most economical and efficient service provider. All transactions are securely recorded on the blockchain, ensuring immutability and data integrity. This dynamic system eliminates the need for manual intervention, allowing the machine to operate independently while maintaining high levels of security.

The benefits of this approach are significant. It enhances cost efficiency by ensuring that the machine always works with the most economical provider (or the best taking into account a set of different criteria). It promotes autonomy by allowing the machine to independently manage its service and payment needs. The blockchain-backed system ensures data integrity and security, while the scalability of the solution supports multiple machines and providers, making it suitable for large-scale industrial environments. This user story highlights how IOTPAY enables IoT devices to autonomously and dynamically manage their transactions in a secure, cost-effective, and scalable manner. This scenario of intelligent IoT devices is becoming more and more realistic with the new wave of AI that creates new needs.

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

4 SOFTWARE ARCHITECTURE

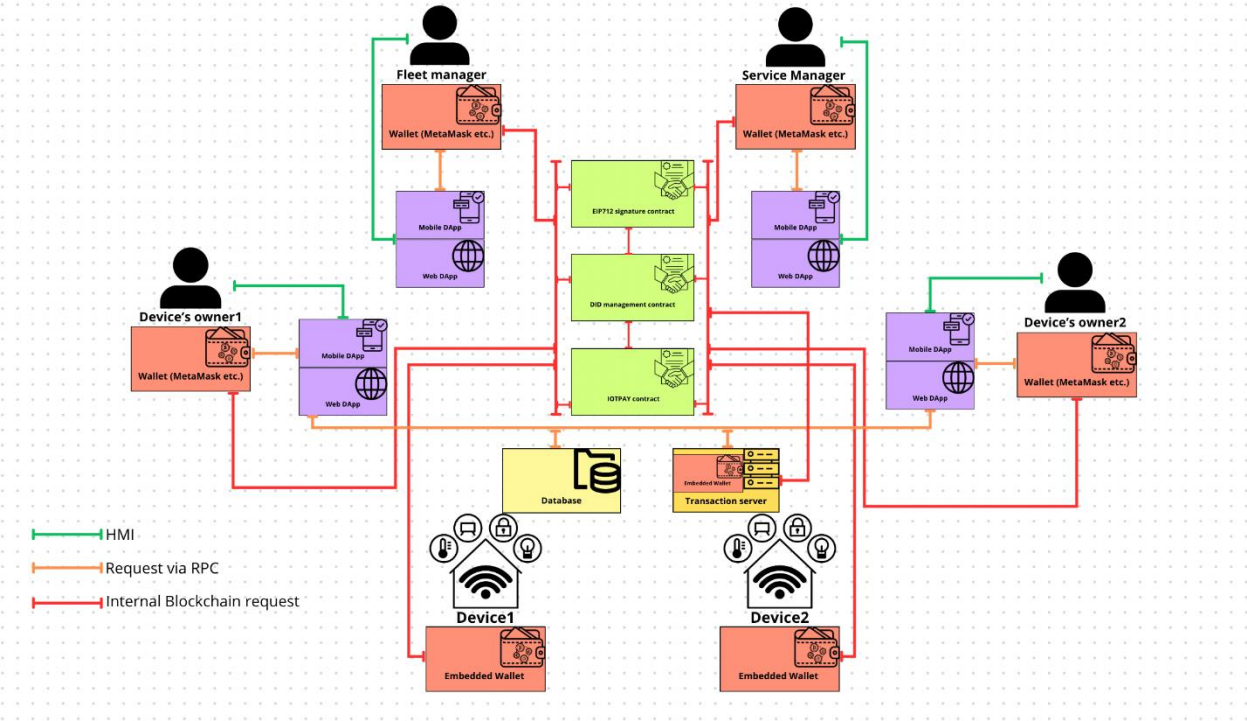

Figure 4 – Software Architecture

The Figure 4 – Software Architecturearchitecture integrates blockchain technology with IoT devices and decentralized identity (DID) management. It focuses on enabling secure interactions between actors (e.g., fleet managers, service managers, device owners) and IoT devices, ensuring seamless management of payments, service credentials, and decentralized identities.

Here are the main software components:

1. Actors and Wallets

- **Fleet Manager, Service Manager, Device Owners:**
  - Use external wallets (e.g., MetaMask) to manage interactions with the system.

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

- Perform signing, credential creation, and payment processing via DApps.

- **IoT Devices (Embedded Wallets):**
  - Devices such as EV chargers or IoT-enabled hardware come with embedded wallets to:
    - Sign blockchain transactions.
    - Authenticate with services via smart contracts.

2. Smart Contracts

Key on-chain components include:

IoT Devices Management Contract

- Manages decentralized identities for IoT devices and actors.
- Supports creation, updating, and revocation of DIDs.

EIP712 Signature Contract

- Implements secure, structured data signing to enable transaction verification.
- Ensures compatibility across wallets and systems.

IOTPAY Contract

- Core payment and service credential management layer.
- Handles:
  - Payment credential creation.
  - Service credential creation.
  - Verification and validation.

3. Middleware and DApps

- **Mobile DApp & Web DApp:**
  - User interfaces for actors to interact with IoT devices and manage credentials or payments.
  - Include RPC requests for blockchain interactions.
- **Transaction Server:**
  - Middleware for efficient transaction handling:
    - Aggregates and queues requests.

- Ensures optimal gas usage by batching transactions.
- Handles blockchain status updates.

4. Off-Chain Storage

- **Database:**
  - Stores metadata and logs linked to on-chain records.
  - Examples include service-level agreements (SLAs), device usage history, and error logs.
- **IPFS or Decentralized Storage:**
  - Holds large or dynamic data (e.g., service metadata, SLA documents).
  - Links this data to on-chain hashes for integrity verification.

Communication Flow

1. **Actor Interaction via DApps:**
   - Fleet managers, service managers, or device owners use mobile or web DApps to initiate operations.
   - Examples:
     - Create a payment credential.
     - Verify service credential validity.
     - Update device ownership.
2. **IoT Device Direct Interaction:**
   - Devices communicate directly with the blockchain using embedded wallets for tasks like signing transactions or verifying credentials.
3. **Blockchain & Middleware:**
   - Smart contracts process the requests submitted via DApps or IoT devices.
   - The transaction server mediates, ensuring efficient blockchain interactions.
4. **Off-Chain Data Linkage:**
   - Off-chain metadata is referenced using content hashes stored in smart contracts.

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

4 PROTOCOL SPECIFICATIONS

Here, the payment credential is not double signed by the service provider. This implies that the currency used is accepted by default by the service provider. There could be a version where a signature from the service provider is necessary; or there could also be a subscription.

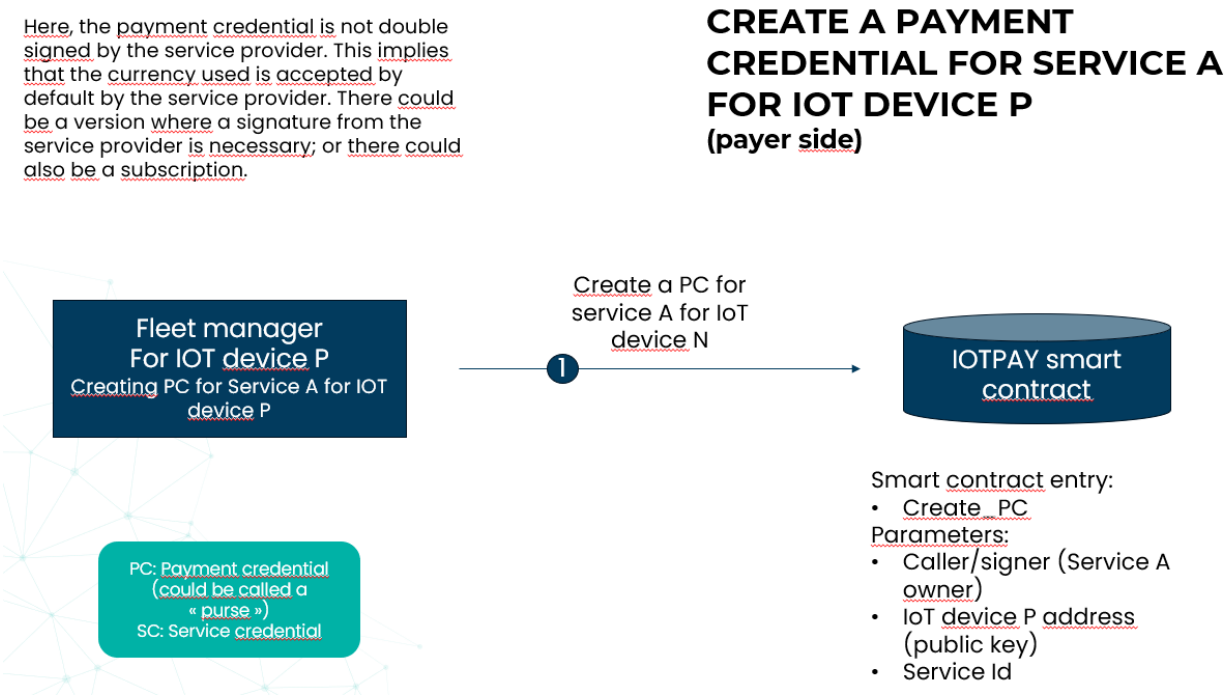

Figure 5 – Payment Credential creation

The Figure 5 – Payment Credential creation outlines the process for creating a payment credential (PC) for an IoT device through the IOTPAY smart contract system. Here are the key points:

1. **Actors:**
    - **Fleet Manager:** Oversees the creation of a payment credential for the IoT device.
    - **IoT Device P:** The target device for which the payment credential is created.
    - **Service A Owner:** Initiates the creation of the payment credential for a specific service.
2. **Smart Contract Functionality:**
    - The smart contract includes an entry point named Create_PC.

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

- Parameters for Create_PC include:
  - The caller/signer, which is the service owner.
  - The IoT device's public key.
  - The service ID.

3. **Payment Credential (PC):**
   - Represents a "purse" that can hold payment data for a specific service.
   - Tied to a specific IoT device and service (here it is so-called Service A).
4. **Process Details:**
   - By default, the payment credential is created without a double signature from the service provider.
   - This implies that the currency is pre-accepted by the service provider.
   - Variations may include requiring a signature from the service provider (double signature) or even signatures from several stakeholders (more than two).
   - or creating a subscription-based model which supports recurring payments tied to the credential.
5. **Wallet Integration:**
   - The IoT device acts as a wallet with a unique public-private key pair.
   - The public key identifies the device and is used to link it to the PC.
   - The PC can is stored on-chain.
6. **Security Requirements :**
   - Inputs securization: Check all inputs in the smart contract to avoid abuse. For example usage of **require** instructions in Solidity.
   - Data Integrity: Use events to log all SC creations for auditing purposes.
7. **Costs concerns**
   - On-Chain Metadata: limit the amount of data stored on-chain to minimize costs.

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

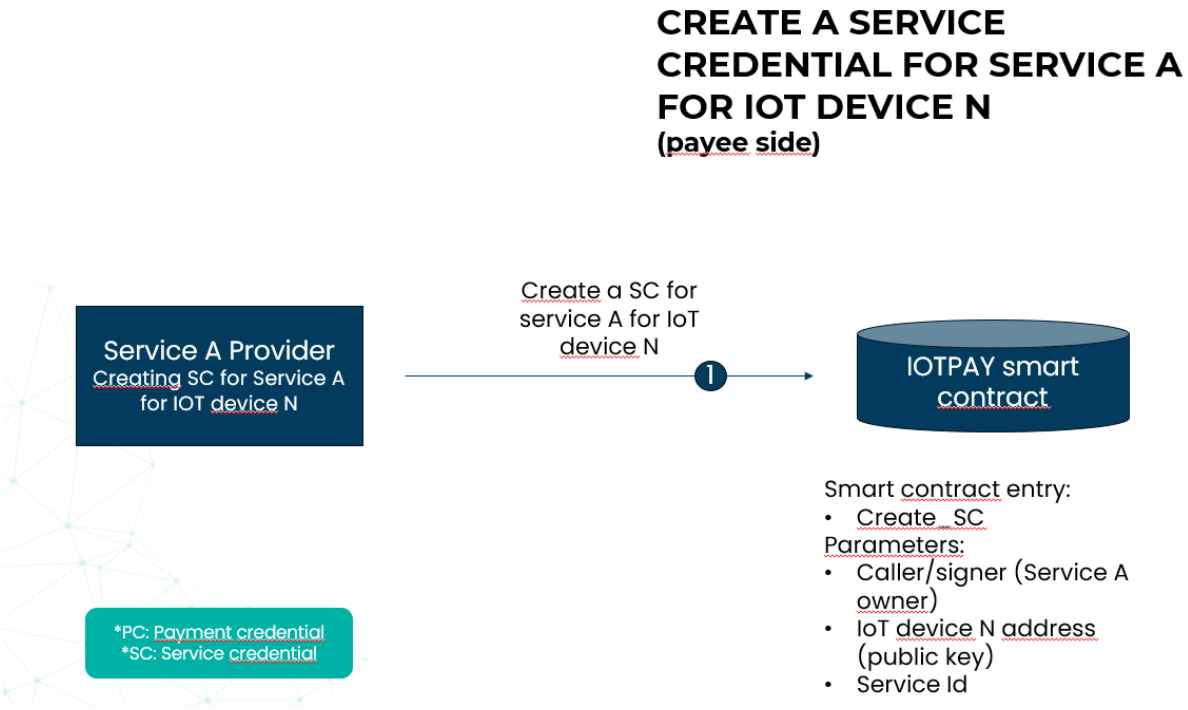

Figure 6 – Service Credential creation

Figure 6 – Service Credential creation details the implementation of creating a **Service Credential (SC)** for IoT devices for the IOTPAY solution:

1. **Service Credential (SC):**
    - The SC acts as a token or certificate, enabling IoT devices to access specific services provided by a Service Provider (or to provide specific services to other IoT devices).
    - It establishes a secure link between the device and the service.
2. **Smart Contract Interaction:**
    - The Create_SC function in the IOTPAY smart contract is used to generate the SC.
    - The function parameters are:
        - **Caller/Signer:** The service provider initiating the credential creation.
        - **IoT Device Address:** The public key of the IoT device being credentialed.
        - **Service ID:** A unique identifier representing the specific service.

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

3. **Purpose:**
    - This credential validates that an IoT device is authorized to interact with a given service (payee-side or payer-side).

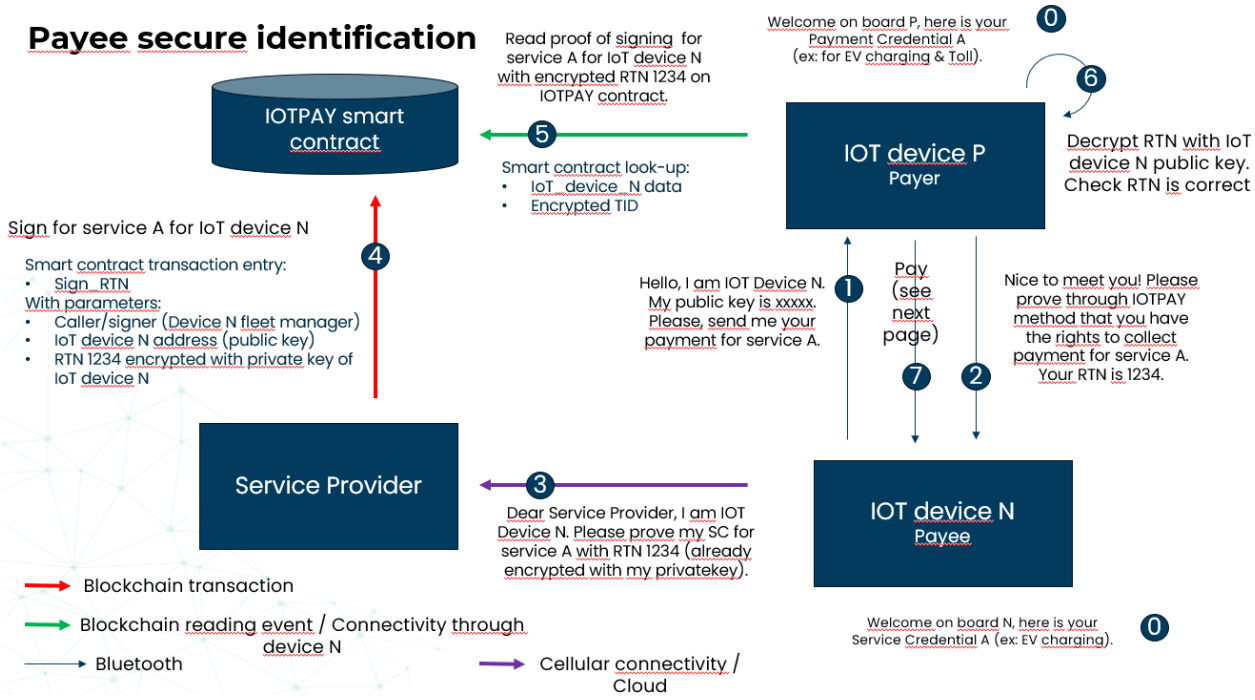

Figure 7 – Payee secure identification

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

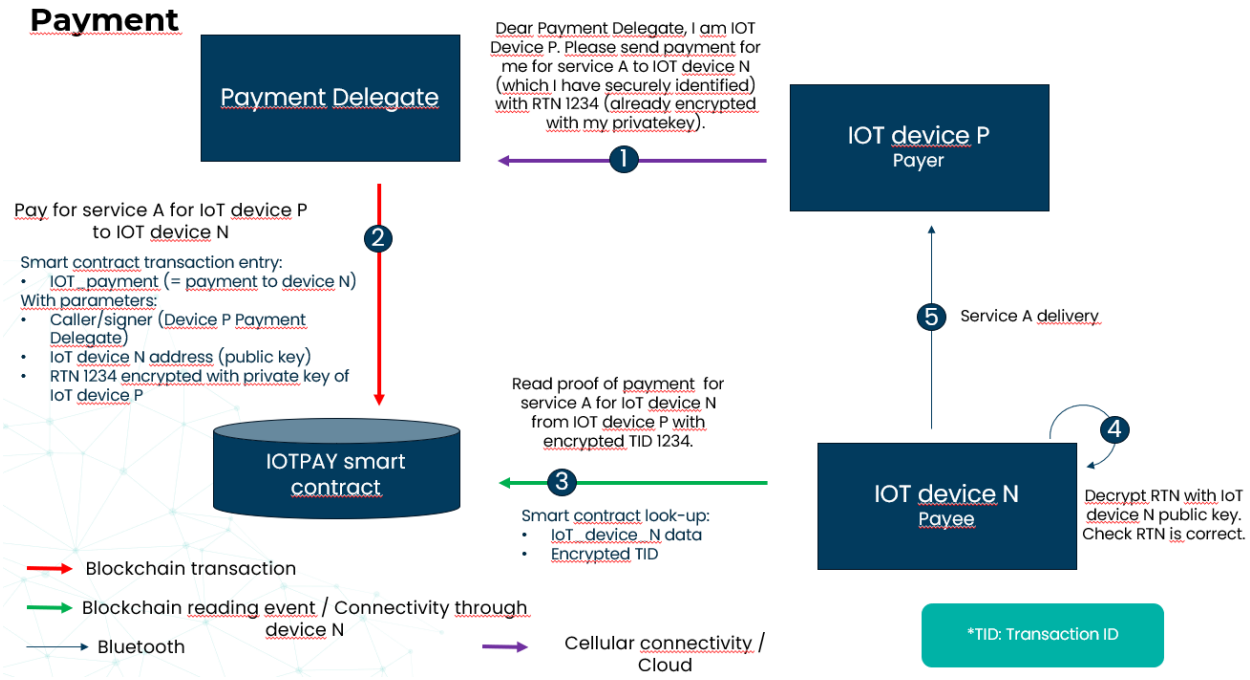

Figure 8 – Payment

5 APPLICATION SPECIFICATIONS

The IOTPAY web application is designed to provide users with a comprehensive and intuitive interface for managing their IoT payment solutions. The key features included are:

- **Dashboard Overview**: Offers a real-time summary of transactions, account balances, and device activities, enabling users to monitor their financial operations at a glance.
- **Transaction Management**: Allows users to view detailed transaction histories, including timestamps, amounts, and involved devices, facilitating easy tracking and reconciliation.
- **Device Management**: Enables users to register, authenticate, and manage their IoT devices within the IOTPAY ecosystem, ensuring secure and authorized participation in transactions.
- **Payment Flow Configuration**: Provides tools for users to create and customize payment authorization ("payment credentials") that define the terms and conditions of automated payments between devices, enhancing flexibility and control.

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

- **Security Settings**: Offers robust security features, including multi-factor authentication and encryption options, to protect user data and ensure the integrity of transactions.
- **Reporting and Analytics**: Generates comprehensive reports and analytics on transaction patterns, device performance, and financial metrics, aiding users in making informed decisions.
- **Integration Support**: Facilitates seamless integration with various IoT platforms and services, providing APIs and documentation to assist developers in connecting their devices to the IOTPAY system.

These features collectively empower users to effectively manage and optimize their IoT payment processes through a user-friendly web interface. They are also described as a DFD (data flow diagram see Figure 9 – Data Flow Diagram).

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

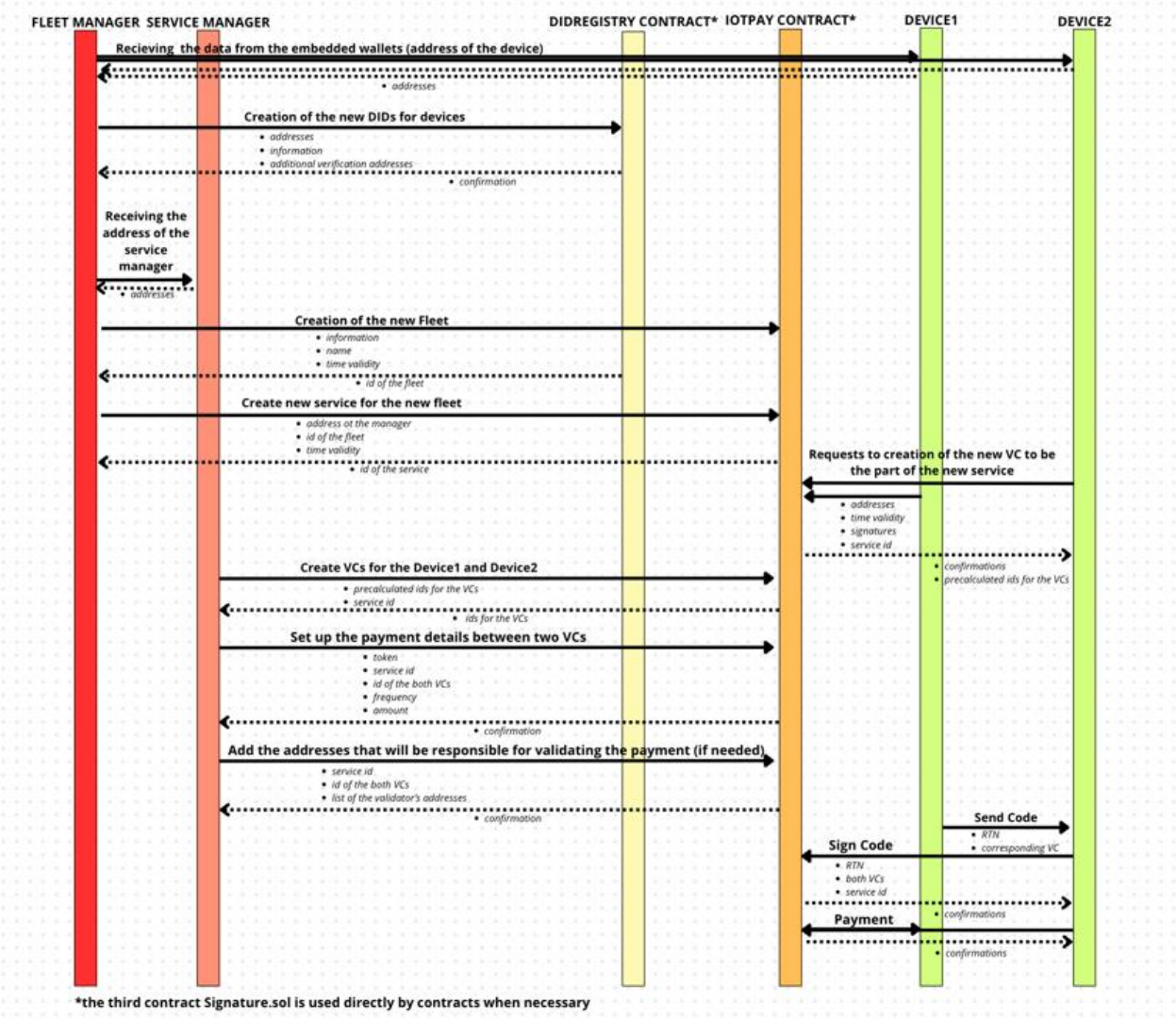

Figure 9 – Data Flow Diagram

The Figure 9 – Data Flow Diagram represents the operational flow of the solution, focusing on device management, fleet creation, and payment validation using decentralized identifiers (DIDs) and verifiable credentials (VCs) implemented as Payment Credentials or Service Credentials. It demonstrates how devices, services, and payment systems interact through smart contracts and verification mechanisms.

Key Components

1. **Devices (Device1 and Device2):**

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

- Represent the IoT devices equipped with embedded wallets that can initiate or receive payments.
- Each device has a unique DID for secure identification.

2. **Fleet Manager:**
   - Manages groups of IoT devices, referred to as "fleets."
   - Responsible for creating new fleets and assigning services to them.
3. **Service Manager:**
   - Manages services associated with fleets, including payment and verification setups.
   - Creates new services for fleets and interacts with contracts for managing service-specific VCs.
4. **DIDRegistry Contract:**
   - A smart contract used for registering and managing device DIDs.
   - Ensures devices have unique and secure decentralized identifiers for interaction.
5. **Signature.sol (Third Contract):**
   - A contract used for signing and verifying transactions and credentials.
   - Supports additional validation processes if needed.

Data Flow Steps

1. **Device Registration:**
   - Devices with embedded wallets send their data (e.g., addresses) to the system.
   - New DIDs are created for these devices and registered in the DIDRegistry Contract.
2. **Fleet Creation:**
   - The Fleet Manager creates a new fleet by grouping devices.
   - The Service Manager assigns services to this fleet, receiving the fleet's unique ID.
3. **Service and Credential Setup:**
   - The Service Manager requests the creation of new VCs for devices in the fleet.
   - VCs contain key information like device IDs, service IDs, and time validity.
4. **Payment Setup:**

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

- Payment details are defined between devices or services via the VCs.
- Validator addresses are added, responsible for verifying the payment.
- Parameters like frequency, amount, and time validity are included in the payment setup.

5. **Transaction Validation:**
   - The payment process involves validators who verify the transaction details (e.g., amounts, addresses, service IDs).
   - Precalculated IDs for VCs are used to ensure secure and quick validation.
6. **Payment Execution:**
   - Tokens are exchanged based on the validated payment information.
   - Confirmation is sent back, indicating successful transaction completion.

This flow diagram highlights the key features of IOTPAY:

- **Decentralized Architecture:** The use of DIDs and VCs ensures the system operates securely without reliance on a centralized authority.
- **Flexible Validation:** The inclusion of validator addresses allows for dynamic verification setups.
- **Scalability:** The Fleet Manager can handle multiple devices and services, making it suitable for large-scale IoT ecosystems.

5.1 Account management

On the main screen (see Figure 10 – Account addresses), we see a list of addresses linked to our account. These include addresses from the connected wallet as well as those manually added by the user via the form below. This form allows the user to assign names to addresses at any time, making it easier to navigate the platform in the future.

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

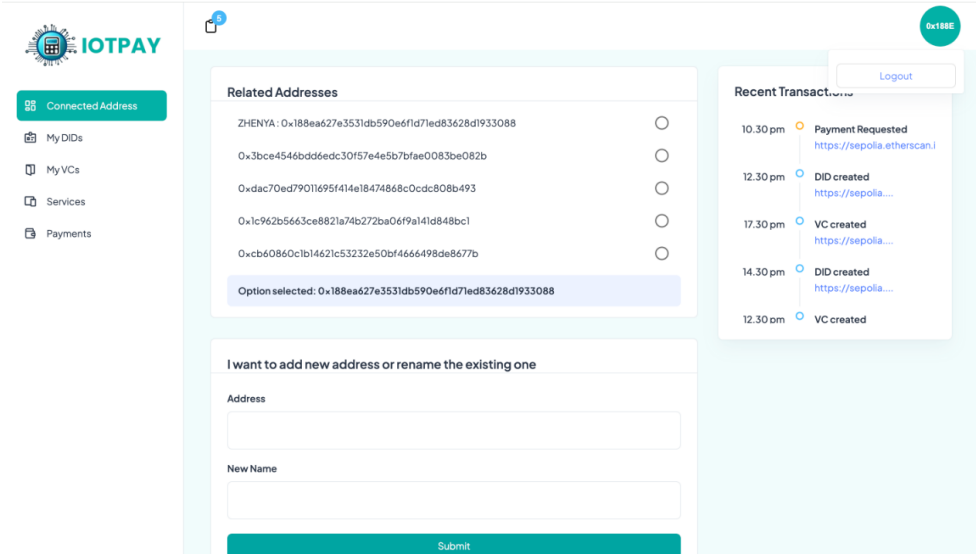

Figure 10 – Account addresses

On the right of Figure 10 – Account addresses, there is also a panel displaying the most recent transactions associated with the selected address from the list.

5.2 Device management

At the top of this page (Figure 11 – IoT device DID management page), there is a list of all IoT devices (and their DIDs) associated with addresses linked to the account. Basic information, such as each DID's status and purpose, is displayed for easy reference.

Clicking on an IoT Device opens a dialog box that allows you to modify the properties of the DID or delete it if needed.

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

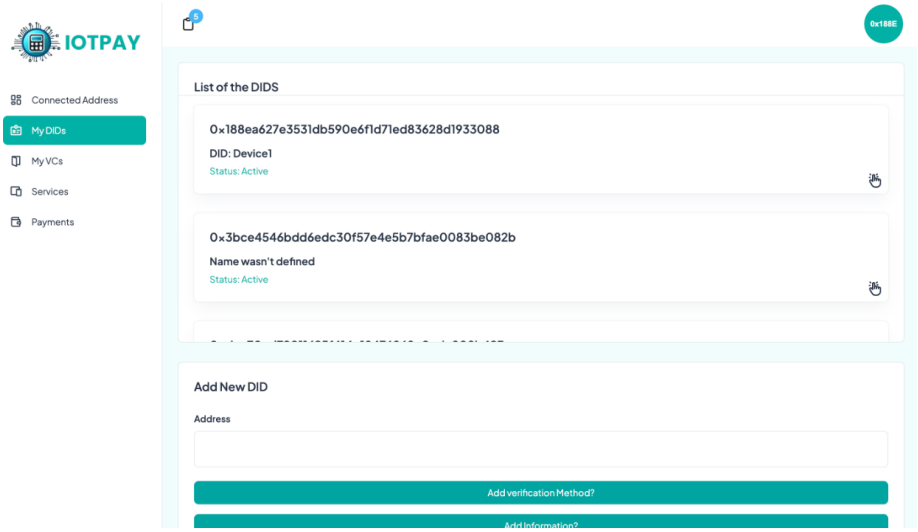

Figure 11 – IoT device DID management page

Below (Figure 12 – IoT device DID creation), there is the form for creating a new DID, where you can specify verification addresses and trusted information sources to validate the DID's purpose. (Note: I plan to add fields for setting time limits on the validity of each DID, as well as an option to rename it.)

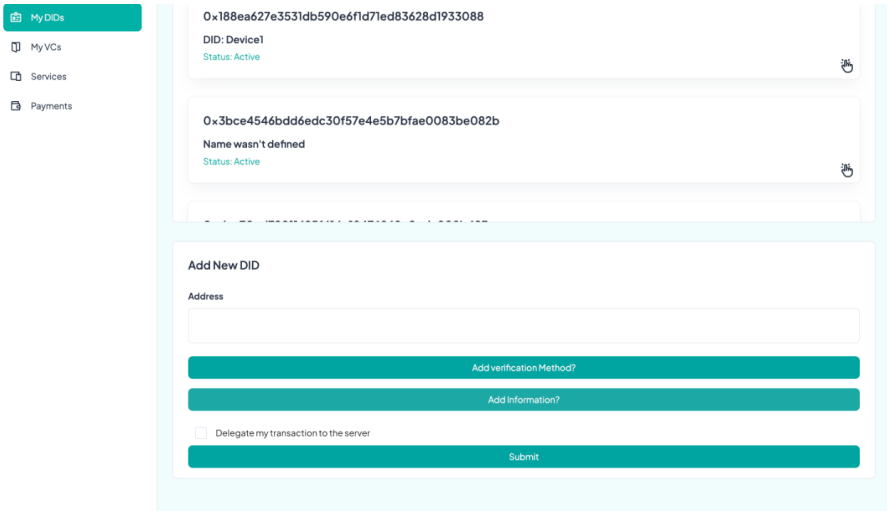

Figure 12 – IoT device DID creation

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

5.3 Credential management

This page (Figure 13 – Credentials management page) displays a list of cards. The first card, when clicked, will eventually prompt the user to create a new VC (currently not implemented, as I am awaiting the smart contracts to better understand their future structure). The remaining cards show a collection of VCs either created by the user or marked for future interaction (all VCs are associated with user-linked addresses, though these addresses don't necessarily belong to the user).

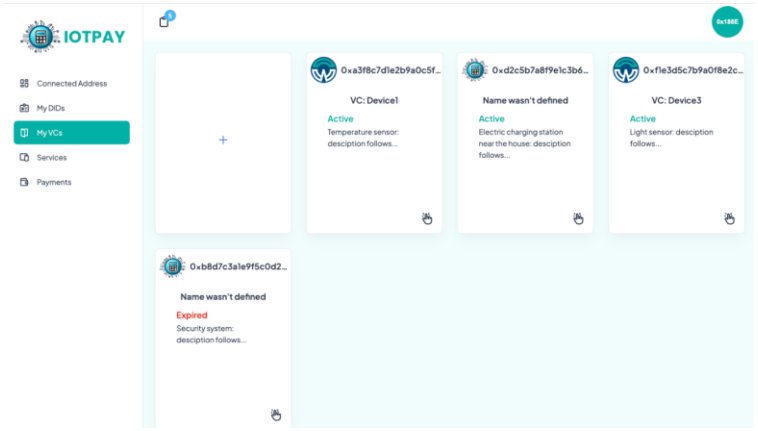

Figure 13 – Credentials management page

Clicking on a VC opens a dialog box (see Figure 14 – Credentials management modal) that shows a list of unpaid payment requests, along with their details. Below, there are options to update the VC's name and description or to delete it.

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

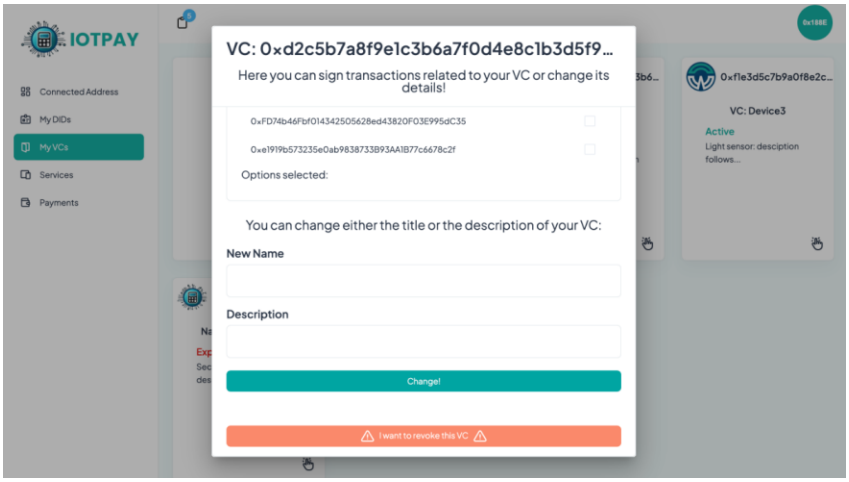

Figure 14 – Credentials management modal

5.4 Services management

On this page (Figure 15 – Services management modal), the user selects a service and a credential from a list, which can also be searched by name or ID.

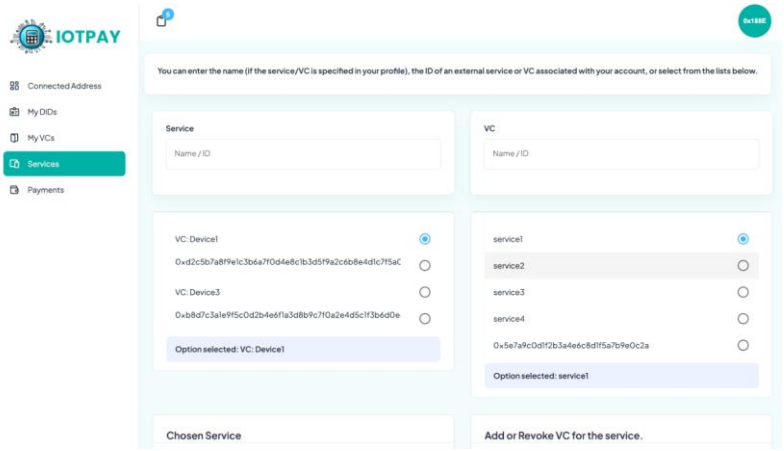

Figure 15 – Services management modal

Below (Figure 16 – Services Information Component), a description of the selected service is provided, detailing its name, ID, manager's address, and a list of included credentials. To the right of this description is a form that allows adding or removing VCs from the service (available only if the user is either the credential owner or the service manager; alternatively, both the manager and VC owner could be required to sign a transaction for removal).

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

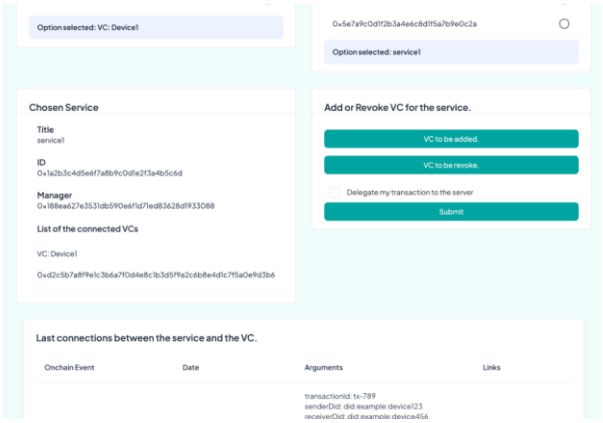

Figure 16 – Services Information Component

At the bottom, a transaction table is displayed (Figure 17 – Transaction history), showing transactions associated with the selected credential and service to help track their interactions.

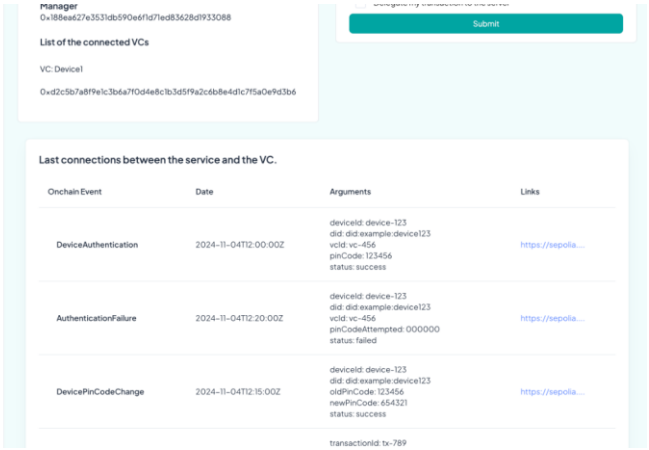

Figure 17 – Transaction history

5.5 Payment management

On this page (Figure 18 – Payment validation), the user is prompted to select a credential and choose whether to view pending or completed payment requests. After making a selection, a list of cards will appear below, each representing an invitation to authorize and pay for services.

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

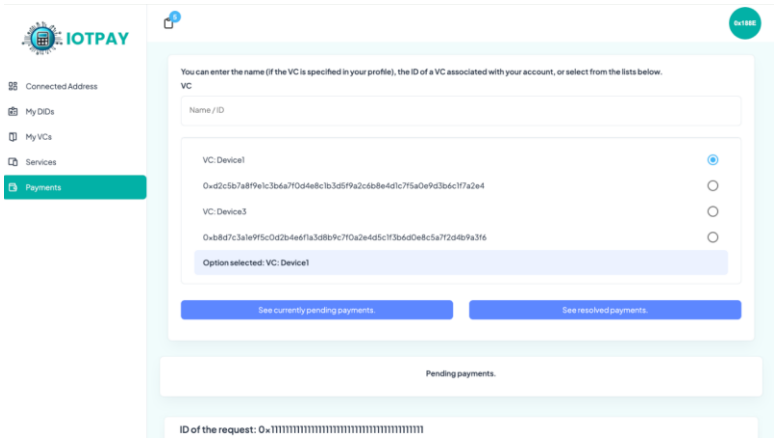

Figure 18 – Payment validation

On this page (Figure 19 – Transaction management), the user is prompted to select a credential and choose whether to view pending or completed payment requests. After making a selection, a list of cards will appear below, each representing an invitation to authorize and pay for services.

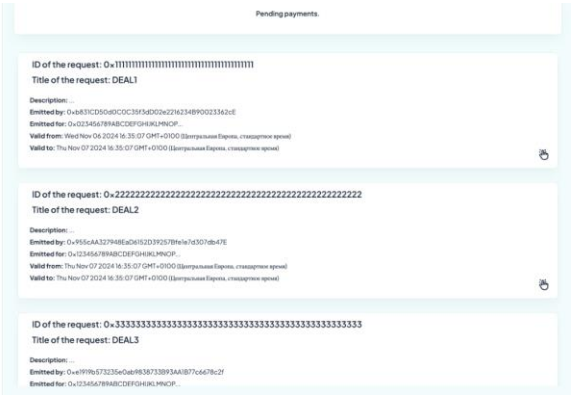

Figure 19 – Transaction management

5.6 User Generated Content

User feedback items

DASHBOARD

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887

1. Recheck how the database updates after each transaction.
2. Validate database data against on-chain information when used.
3. Ensure the frontend reflects changes when the database is updated.
4. Hide signValid fields for adding devices (except for pending cases).
5. Consider improving the list of graphs (visualization) for better user data representation.
6. Remove nonce fields from all forms and service contracts (retrieve it for a specific address directly from the smart contract).
7. Fix the wallet connection in MemberApp.
8. Implement event monitoring to notify the manager/devices about the need to confirm the creation of a new VC, sign the code, validate the payment, and make the payment.

CONTRACTS

1. Implement a function to delete a fleet.
2. Create a function to change the payment status.
3. Enable payment with a flexible amount, up to the maximum limit.
4. Add a description field to the signature of the code to indicate its purpose.
5. Develop a function that returns an array of paymentKeys by vcId (for the VC graph).
6. Review the data representation during transaction signing; consider changing data types where necessary for better user convenience.

DEVICE'S FUNCTIONALITY

! In all data types, nonce is currently included. Later, it will be removed and we will insert it automatically instead of requiring it from the user. !

- First, the manager creates a DID on the platform for the embedded wallet on the device.
- The manager creates a fleet.
- The manager creates one or more services.
- Then the **device** must sign the transaction to request a VC:

This project has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101092887