THE JBBA

PEER-REVIEWED RESEARCH

OPEN ACCESS
ISSN Online: 2516-3957
ISSN Print: 2516-3949
https://doi.org/10.31585/jbba-7-1-(2)2024

A Blockchain-Based, Smart Contract and IoT-Enabled Recycling System

Manaf Zghaibeh
Department of Electrical and Computer Engineering, Dhofar University, Salalah, Oman

Correspondence: mzghaibeh@du.edu.om
Received: 12 July 2023 **Accepted**: 15 September 2023 **Published**: 21 October 2023

Abstract
The current state of recycling systems is marked by significant impediments to their efficacy. A lack of transparency often pervades these systems, which may result in an increased likelihood of fraudulent and corrupt activity. Additionally, traceability pertaining to recycled materials frequently proves inadequate. Together, these inefficiencies in the collection and processing of recyclables can lead to higher costs and environmental impact. Furthermore, low incentives may deter individuals and businesses from participating in recycling initiatives. Certain recycling systems may also suffer from limited compatibility with specific materials, further reducing their effectiveness. To address these challenges, we propose a permissioned Ethereum blockchain-based system that aims to incentivise and encourage recycling practices in a transparent and secure manner. The platform's modular and multi-layered design makes it adaptable to various recycling scenarios, allowing it to handle diverse types of recyclable materials. Automated and streamlined recycling processes are achieved through the use of smart contracts. The proposed system offers a secure, transparent, and efficient platform for the management of recycling processes, promoting environmentally responsible behaviour towards a circular economy. Potential applications for the system include waste disposal and recycling management for smart cities, waste management for organisations, and tracking and management of operations for recycling companies. The platform is highly versatile and can accommodate various use cases in the recycling industry, including those involving traceable and untraceable materials, as well as individual and corporate use cases.

Keywords: *Blockchain, Sustainability, IoT, Recycling, Smart Contracts, DPoS*

JEL Classifications: *D82*

1. Introduction

Recycling plays a pivotal role in mitigating the carbon footprint and combating the environmental repercussions associated with single-use waste, contributing significantly to the principles of the circular economy [1]. However, the efficacy of voluntary recycling programmes at a large scale is often questioned when compared to mandatory initiatives [2]. The reliance on individual adherence to consistent and proper recycling practices in voluntary programmes has proven challenging, resulting in contamination and compromised quality of recycled materials [3]. To address these concerns, the enforcement of recycling policies and regulations becomes imperative. Mandatory recycling programmes, bolstered by effective enforcement mechanisms, establish individual responsibility for appropriate waste recycling [4, 5]. This approach holds the potential to elevate recycling rates and foster the production of superior-quality recycled materials, thereby diminishing the demand for virgin resources and promoting sustainable production practices [6, 7, 8].

Numerous studies have explored the integration of technologies to optimise waste collection, sorting, and recycling processes. For instance, IoT sensors have been used to collect data on waste generation, predict waste amounts, and optimise waste bin collection processes, as demonstrated in [9, 10]. In [9], the authors designed a system that uses IoT sensors to collect data on waste generation and designed an algorithm to predict the amount of waste generated. The system also provided information on the location and capacity of waste bins to optimise the collection process. The authors concluded that their system could reduce the time and cost of waste collection while also promoting recycling by providing insights into waste composition. Similar to [9], [10] evaluated the performance of a smart waste management system in a university campus in Taiwan. The system used RFID technology to track the movement of waste bins and sensors to determine the fill level of each bin. The study found that the smart waste management system improved waste collection efficiency and reduced the overall collection frequency. In the realm of blockchain, several studies discussed the implementation of blockchain to recycle e-waste in particular [11, 12, 13, 14]. [11] specifically emphasised the effectiveness of deploying blockchain technology effectively in order to improve the recycling rate of waste electronics and building trust in consumers. [12] explored the capabilities of a blockchain system to track products and analysed different aspects of costs associated with implementing blockchains for solid waste management and costs spent by existing waste management companies to adapt to the blockchain platform.

The systems in [13, 14, 15] suggest blockchain-based e-waste tracking systems for smart cities. The main goal of these systems is to address the issues associated with e-waste management specifically. To achieve this, the proposed solutions combine the use of RFID tags and blockchain technology to monitor and track e-waste throughout its lifecycle, from generation to disposal or recycling. However, these systems only track e-waste that are originally equipped with RFIDs and do not provide incentive and penalising mechanisms to promote recycling and penalising. Furthermore, they require a significant investment in infrastructure for such specific purpose, and their scope and scale are limited with predefined functionalities.

In this paper, we introduce a permissioned Ethereum blockchain-based platform that aims to encourage and incentivise recycling through a transparent and secure system. It provides a digital platform where clients can track their purchase and recycle activities to realise their impact on the environment. This ensures that all parties involved have access to accurate and verified data. The system's main goal is to establish a sustainable ecosystem that incentivises and fosters responsible behaviour towards the environment. The platform achieves this by employing a tracking system for the acquisition of disposable items. By assigning unique identifiers to recyclable items and utilising scanning technology, it tracks the entire lifecycle of these items, starting from their production to their eventual disposal. This valuable information enables authorities to identify products that are not being recycled and determine their final destination. Such insights can inform targeted interventions and policies to improve recycling rates and minimise the negative impacts of improper waste disposal [16]. This method would also help to increase accountability among manufacturers, distributors, and consumers. By having an accurate record of the products that are not being recycled, authorities could penalise or fine organisations or individuals who are not properly disposing of their waste [17]. In contrast to previous work on blockchain, our system offers a comprehensive waste management solution that can handle all types of waste, including both recyclable and organic materials. Through its smart contracts capabilities, the platform ensures the secure and efficient tracking of waste from the point of purchase to its disposal or recycling. Unlike other blockchain-based waste management systems, such as those discussed in [14, 18], the proposed system has the flexibility to adapt to different types of waste and use cases, making it a highly versatile and scalable solution for promoting responsible waste management practices. Furthermore, the potential uses of the system can be expanded to accommodate a wide range of waste management scenarios beyond its initial scope, enhancing its utility and value to users. The main contributions of this work can be summarised as follows:

- Incentivising recycling system: Unlike other blockchain systems that only track the recycling of specific material, our proposed system is the only one that functions as an all-encompassing circular ecosystem which employs blockchain technology to incentivise and promote proper recycling practices [1]. Through the utilisation of blockchain, users' purchasing and recycling activities are meticulously recorded and tracked.
- Modularity: The system's modular design enables it to be adaptable and flexible to different recycling scenarios, thanks to its multi-layered and multi-tiered structure. This modularity allows for easy integration with existing recycling infrastructures and can be customised to suit the specific needs of individuals or businesses. Additionally, the system's modular approach enables the platform to evolve and expand over time to include new features and functionalities as required, making it a sustainable solution for waste management.
- Ethereum-based: The Ethereum network is a popular blockchain platform that supports smart contracts and decentralised applications DApps. It is known for its security features and its ability to handle large amounts of data and transactions. Our system leverages the Ethereum network to ensure the security and transparency of its waste management platform, enabling users to track and manage their recycling activities efficiently.
- Applicability: Unlike other blockchain waste management systems, the applicability of this proposed system is not limited to a specific type of recyclable items, and it is designed to accommodate both traceable and untraceable items. This feature enhances the system's versatility and enables its integration into different recycling scenarios, thereby offering a highly adaptable and scalable solution.
- Use cases: The modular design of the system allows for the implementation of different use cases targeting various recycling scenarios. For example, it can track recyclable materials in households, businesses, and public places, as well as incentivise users through rewards for responsible recycling behaviour. Additionally, it can also be used to track the recycling of hazardous materials such as batteries and electronic waste. This versatility in application makes the system a valuable tool in promoting responsible behaviour towards the environment and reducing the negative impact of waste on our planet.

2. The System

The system is structured into distinct layers, each housing elements that possess unique roles and responsibilities. These elements are identified and categorised by the system based on their designated addresses (Figure 1):

2.1 Layers

The initial layer in the system is designated as the **Control Layer**, serving as the foremost authoritative entity within the system. The primary function of this layer is to regulate access to the blockchain through the process of sanctioning new nodes and admitting new clients based on their national identification. The term "node" refers to a point of sale, encompassing facilities such as hypermarkets, supermarkets, grocery stores, and vending machines that provide recyclable products for sale. Within our system, the assignment of responsibilities is formed into three distinct levels, each categorised based on the capabilities and resources possessed by the nodes. Nodes with greater capabilities, including hypermarkets and supermarkets, occupy level 1 and maintain a comprehensive copy of the ledger. The sanctioning process for level 1 nodes entails an on-site, off-chain, bureaucratic procedure, wherein the node is formally recognised as a licensed participant in the system and assigned a unique address within the blockchain. Additionally, nodes at this level possess the authority to authorise nodes in the subsequent level, in addition to clients and IoT terminals like vending machines, recycling depots, and bins.

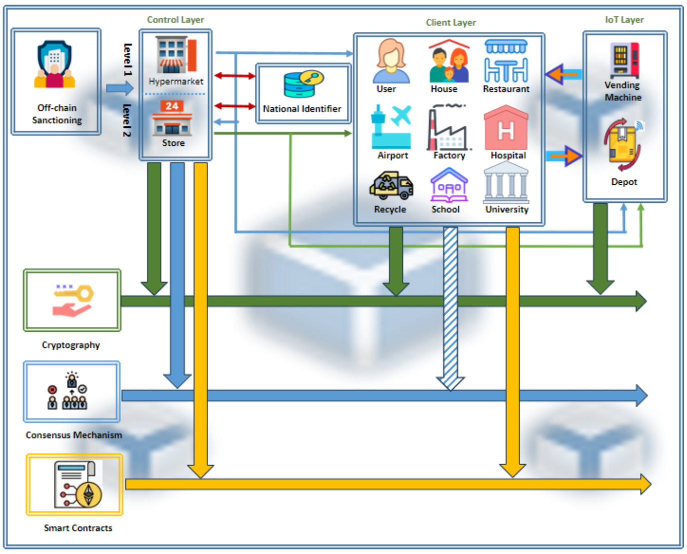

Figure 1. Abstraction of the proposed system.

Level 2, situated within the control layer, comprises nodes with relatively fewer resources, such as small markets and retail shops. Similar to level 1 nodes, those in level 2 retain the entire ledger; however, their sanctioning privileges are limited to clients and IoT terminals at the subsequent level. Nodes in this level do not have the authority to sanction other nodes. This arrangement of distributed functionality among the nodes allows for a more efficient and practical use of resources within the retail blockchain system. Furthermore, each node has a unique private cryptographic key that acts as its identification in the system, used to generate its public cryptographic keys that are shared with the system in a secure manner.

To enable automated and precise monitoring, the system utilises IoT devices to capture real-time data on product acquisition and disposal. This data is securely stored on the blockchain, ensuring transparency and accountability throughout the purchasing and recycling process. Accordingly, the **IoT Terminals Layer** of the system hosts authorised IoT-enabled devices, such as vending machines and recycling depots, which play a crucial role in scanning and documenting the purchase and disposal of products. They possess no sanctioning capabilities and do not hold the entire ledger; rather, such terminals only require access in writing mode to add entries to the client's records in the ledger. To ensure the authenticity and integrity of the IoT devices within the network, a registration process is implemented in the control layer: Before an IoT device can be recognised as a sanctioned apparatus, it must undergo registration at the control layer and receive a unique system identity. This registration process establishes a trusted relationship between the IoT device and the platform, ensuring that only authorised devices contribute to the purchasing and recycling process.

The third layer in our system is the **Client Layer**. A client refers to an individual or organisation that interacts with the platform to participate in purchasing and recycling activities such as individuals, households, restaurants, airports, factories, hospitals, schools, universities, and government and private offices. The admission of new clients to the system is facilitated by the generations of addresses, which are based on government-issued national or tax identifications. Levels 1 and 2 nodes of the control layer in the system are responsible for coordinating with other non-system agencies that maintain the tax or the national identifier database. During the registration process, an individual submits an application with their unique national identifier, which must then be validated by the relevant government agency.

Clients within the system are identified by their unique addresses in the blockchain. Each client is assigned a 2-of-2 multisignature address, which ensures the secure storage of their transactions. Clients do not have exclusive control over their records in the blockchain. Instead, they can collaboratively add new transactions to their records through their associated node or terminal, which is responsible for providing the product or handling the recycling process.

After creating their addresses, clients can access information about recycling locations and events, record and track their purchasing and recycling activities, and earn credits for their contributions to environmental sustainability. They access the system through their wallets.

The wallet, available at trusted locations such as nodes or affiliated websites, provides several functionalities, including key storage, request initiation, and record viewing. It is conceptualised as a software application that is installed on the client's mobile device or terminal. This application holds the private cryptographic key that serves as the client's identification within the blockchain network. To maintain security, the private key must be kept confidential and not shared with unauthorised individuals. In order to provide access to the client's records within the blockchain, the wallet generates a public cryptographic key, derived from the private key. This public key is then transmitted to the node, granting it permission to access the client's records. The lifespan of the public key is determined by a time limit set within the wallet system, which can be altered based on the client's specifications and expires after a predetermined interval.

Recycling companies are also located in the client layer. Those are clients that collect, process, and sell recyclable materials, such as paper, plastic, glass, and metal, to manufacturers that use these materials to make new products. They are sanctioned into the system similar to regular clients by nodes in levels 1 and 2. Upon joining the system, a recycling company will be assigned a "Credit" ledger based on its recycling capacity and collection effort. The credits, or tokens, in this ledger are used to pay other entities for the amount of untraceable recyclables they generate and require special collection and treatment. The credit ledger for each company in the system is reviewed regularly and increased or decreased based on the company's recycling performance. Ultimately, a blockchain constitutes the inclusive layers aforementioned, excluding the off-chain components, and serves as the repository for clients' factual records and transactions. Specifically, this private blockchain serves as the pivotal depository for all client records within the system. Access to this decentralised ledger is conferred upon all authorised nodes ensuring its widespread accessibility. The blockchain operates on the premise of replication, thereby safeguarding previous records against tampering while permitting read-and-write operations with the client's explicit consent. These records are organised into two distinct stacks: one for confirmed purchased items and another for confirmed disposed items.

2.2 Addresses

Entities within the system possess distinct privileges and responsibilities based on their designated address class. Level 1 nodes in the control layer are assigned *Class 1* addresses, granting them the authority to approve new nodes and clients, as well as maintain the global blockchain. Level 2 nodes are assigned *Class 2* addresses, enabling them to admit clients, participate in transaction verification and approval, and maintain a complete copy of the blockchain. However, they lack the authorisation to sanction new nodes. On the other hand, IoT terminals such as vending machines and recycling depots are assigned *Class 3* addresses, allowing them to engage in transaction verification and approval only. Furthermore, clients, designated with *Class 4* addresses, do not possess sanctioning privileges or participate in the consensus mechanism. However, they can initiate and execute smart contracts and access their own records. Lastly, *Class 5* addresses represent n-of-m multisignature addresses exclusively reserved for smart contracts. These addresses are initiated by clients and triggered by nodes within the system.

The process of generating cryptographic addresses, also known as private-public key generation or asymmetric cryptography, involves generating a private key that must be stored in secrecy to ensure data security. Asymmetric cryptography is widely used in blockchain technology to ensure the authenticity and confidentiality of transactions [19, 20]. In the process of private-public key generation or asymmetric cryptography, a private key is randomly generated and should be kept secret by its owner. This key is used to create digital signatures, which are required for proving ownership of records. Applying the private key to a transaction generates a numerical signature, and it is also used to decrypt messages that were encrypted with its public key. The public key is derived from the private key and is used by other entities to encrypt messages addressed to the key's owner. Transactions in the network can be directed to the client's public key, yet for added security, it is recommended to generate addresses from the public key using a hashing algorithm instead of using the public key itself as an address [21, 22].

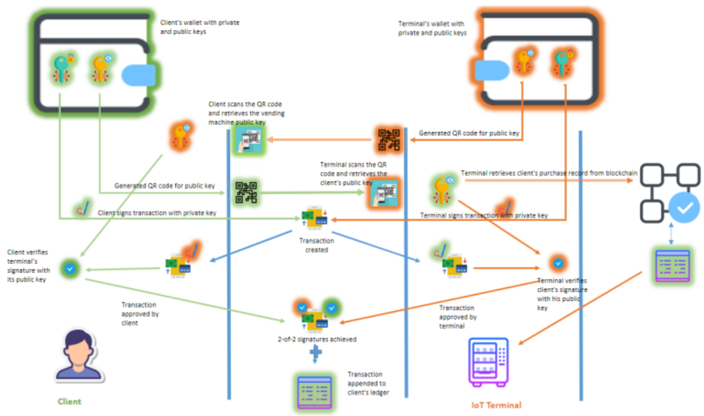

Figure 2. Sequence diagram illustrating the 2-of-2 multisignature scheme as a record is being appended to the client's ledger.

2.3 Multisignature

The use of multisignature schemes in blockchain technology provides an additional layer of security and accountability to transactions. A multisignature scheme is a security mechanism used in blockchain technology that requires the approval of multiple parties before a transaction can be validated [23]. In a traditional transaction, a singular party initiates the transaction, affixes their digital signature using the corresponding private key, and subsequently broadcasts the transaction into the blockchain. However, in a multisignature scheme, multiple parties are required to sign the transaction before it can be verified and added to the blockchain. The most common multisignature scheme used in blockchain is the *n-of-m* scheme, where *n* represents the number of signatures required to validate a transaction, and *m* represents the total number of parties involved.

All transactions within our system, including the addition of purchase or recycle records, are completed through the utilisation of *n-m* multisignatures. The implementation of multisignature ensures that every transaction necessitates approval from a minimum of two parties, thereby augmenting the system's security and transparency – for instance, when a client is purchasing a bottle of water from a vending machine, which serves as an IoT terminal in our system (Figure 2). The transaction process begins with both the client and the machine exchanging their public keys which are facilitated as scannable QR codes. Subsequently, using the client's public key, the machine retrieves their records from the blockchain and verifies his eligibility to complete the purchase process. If the client's purchase record is full demonstrating negligent recycling behaviour, the transaction will be reverted (Figure 3). Conversely, after verifying the records, both the client and the vending machine employ their private keys to sign the transaction that appends the identifier of the bottle to the client's purchase record in the blockchain. The signatures of both parties are verified by testing them against the public keys they retrieved from each other in the first stage. Finally, the transaction is emitted into the network for approval as detailed in Algorithm 1. Figure 4 illustrates the successful execution of the VendingMachine smart contract.

Figure 3. Remix IDE output for VendingMachine contract in Algorithm 1: The transaction has been reverted due to reaching a maximum level of purchased items.

Figure 4. Remix IDE output for VendingMachine contract in Algorithm 1: The transaction mined and executed successfully.

2.4 Blocks

The system's blockchain consists of blocks, wherein each block contains a verified list of transactions executed within the network. These transactions are denoted by *2-of-m* multisignature records assigned to individual clients for recyclable items. The blockchain promotes accountability and sustainability by tracking both purchased and recycled items. Hence, as mentioned earlier, in our system every client has two distinct records: one for purchased items and the other for recycled items. To illustrate, consider the scenario where a client wishes to recycle a previously purchased item. He disposes the item at the recycling depot, where it is scanned for its unique identifier and signed (Algorithm 2). The depot then searches the client's purchase record to check if he had previously purchased the item. If a match is found, the recycling depot initiates a transaction to remove the entry from the client's purchase record. Otherwise, the depot initiates a transaction to add the item to the client's recycled items. Either of the signed transactions is then broadcasted to the network for verification and eventual inclusion in the client's record in the blockchain (Figure 5).

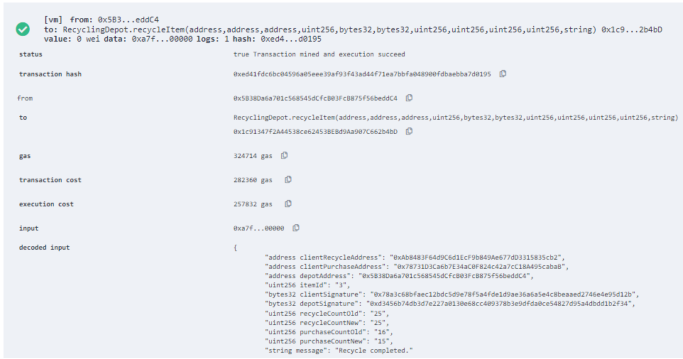

Figure 5. Remix IDE output for RecyclingDepot smart contract in Algorithm 2: The transaction mined and executed successfully with updated recycling and purchasing records.

2.5 Consensus mechanism

Given the nature of our system as an ecosystem that tracks individual transactions related to recycling, a consensus mechanism that prioritises speed and scalability over security is more suitable. This is because the proposed system is likely to have a large number of transactions that need to be processed quickly and efficiently. Hence, a consensus mechanism like Delegated Proof of Stake (DPoS) elevates as a good fit for this system. DPoS is faster and more scalable than both Proof of Work (PoW) and Proof of Stake (PoS) and is more suitable for ecosystems that require high transaction throughput. Additionally, DPoS is more energy-efficient than PoW, which is a significant consideration in a system that tracks individuals' carbon footprint.

The consensus algorithm is utilised to validate all *n-of-m* transactions carried out within the system by clients, nodes, and terminals. In our system, nodes at levels 1 and 2, along with clients of substantial capacity, super-clients, such as schools, hospitals, and universities, actively participate in the consensus mechanism. Conversely, smaller entities within the system, such as regular users and households, do not engage in the consensus mechanism. The exclusion of smaller entities, such as regular users and households, from the consensus mechanism is a deliberate design choice to ensure scalability and efficiency. By limiting the participation to nodes and clients with substantial capacity, the system can maintain a manageable number of delegates while still benefiting from their expertise and resources.

In our system, the delegate node selection process is automated and endeavours to elect nodes with transparency and efficiency as its primary objectives. By incorporating specific criteria and constraints, the underlying algorithm systematically identifies a subset of delegate nodes that assume pivotal roles within the blockchain consensus mechanism. The selection process entails a comprehensive evaluation of multiple criteria, encompassing factors such as node's uptime, accumulated tokens from previous processing, node's level, processing capacity, and the location of the node in the area. Each node's performance in these areas is meticulously assessed, and a reward function is employed to quantitatively gauge their overall suitability as delegates. For a system of N nodes participating in the consensus mechanism, we set the following:

$U(n)$: Uptime of node n for $n \in N$.

$T(n)$: Tokens collected from previous processing for $n \in N$.

$L(n)$: Node level (1 or 2) or super-client for $n \in N$.

$C(n)$: Processing capacity of node n for $n \in N$.

$S(n)$: Node location (sector) for $n \in N$.

$D(n_1, n_2)$: Euclidean distance between nodes n_1 and n_2 for $n_1, n_2 \in N$

Where the objective function is to maximise combination of factors such as node uptime, tokens collected, node level, processing capacity, and distance between nodes:

$$\sum_{n \in N} \big(U(n).X(n)\big) + \sum_{n \in N} \big(T(n).X(n)\big) + \sum_{n \in N} \big(L(n).X(n)\big) + \sum_{n \in N} \big(C(n).X(n)\big) - \sum_{n_1, n_2 \in N} \big(D(n_1, n_2).X(n_1).X(n_2)\big)$$

expecting the output of S, the set of selected delegate nodes, where $X(n)$ is the binary decision variable indicating if node n is selected as a delegate for $n \in N$, subject to the constraints that include selecting a desired number of delegates, level restrictions, fair selection from sectors, and prioritising processing capacity: $\sum_{n \in N} X(n) = k$ where k is the desired number of delegates to be selected, $X(n) \leq L(n)$ for all $n \in N$, to ensure that level 2 nodes or super-clients are selected only if necessary, $\sum_{n \in N, S(n)=s} X(n) \geq (1 - \varepsilon)(\frac{k}{s})$ for all s, where s is the number of sectors in the geographic area where the nodes are located, and ε is a small tolerance value, to ensure fair selection from sectors, $\sum_{n \in N} (C(n).X(n)) \geq \theta . \sum_{n \in N} \big(T(n).X(n)\big)$ where θ is a trade-off parameter, to prioritise processing capacity over tokens collected, and $X(n) \in \{0, 1\}$ for all $n \in N$

3. Use cases

3.1 The case of businesses

A proprietor of a downtown restaurant that generates a significant amount of recyclable items creates a smart contract to facilitate the collection of these recyclables in exchange for a fee.[1] The smart contract encompasses the terms of the agreement between the restaurant and recycling companies, specifying the types and amount of recyclables to be collected and the corresponding service fee. Upon creation, the smart contract is disseminated across the network, allowing interested parties to participate (Algorithm 3). The execution of the smart contract occurs when at least one recycling company acknowledges and accepts the contractual terms and conditions. The involved parties then proceed to endorse the transaction by signing the associated 2-of-m multisignature address. During the collection process, the recycling company scans the deposited recyclables and ascertains their quantities and values. Upon completion of the recycling process and the mutual agreement on the value of the collected recyclables, the parties affix their signatures to the transaction, which is subsequently submitted for approval on the blockchain. Consequently, the restaurant's "purchase record" is adjusted and reduced to account for the items that have been acquired through recycling, and the company's "recycling record" is updated to indicate the items they have received from the restaurant. Figure 6 shows an excerpt of RecyclingContract smart contract output. It shows the success of executing the function executeTransaction presented in Algorithm 3. The function executes after the recycling company agrees to collect 200 items from the client for a fee of 100 according to the function acceptAgreement. In conclusion, the result of recycling the 200 items is reflected in both the client's purchase record and the company's recycle record.

Figure 6. Remix IDE output for the executeTransaction function in the RecyclingContract contract in Algorithm 3.

[1] Implementing the generation of smart contracts through a GUI interface within the user's wallet offers a user-friendly approach to creating blockchain-based agreements. This intuitive method streamlines the process by integrating contract creation directly into the wallet interface. Users can interact with the wallet's graphical tools to design, configure, and deploy smart contracts without the need for extensive coding knowledge.

3.2 The case of untraceable recyclables

Untraceable recyclables are those types of waste that are difficult to track and monitor throughout the recycling process due to a variety of reasons such as having a longer lifespan and multiple uses beyond their initial purchase. Some common examples of untraceable recyclables include paper, bulk plastic, bulk metal, and construction and demolition waste [24]. These types of untraceable waste present significant challenges for recycling programmes and require innovative solutions to incentivise their disposal and recycling. For example, tracking the recycling of paper items poses a significant challenge compared to other recyclable products. The primary reason is that paper has a longer lifespan and multiple uses beyond its initial purchase, making it difficult to track its disposal and recycling. Some of the paper purchased may be archived or stored for long periods, rendering it impossible to trace its recycling journey. For instance, while plastic bottles have unique identifiers that make tracking their purchase and recycling records relatively easy, items such as A4 paper blocks lack this feature. To overcome this challenge, incentives should be created to encourage the recycling of paper products. One such incentive is rewarding individuals and entities based on the volume or weightage of waste they recycle by crediting their “recycling records.” As an illustration, a company that wishes to recycle considerable amount of paper waste generates a 2-of-m smart contract in the network requesting recycling companies to submit their bids for collecting the paper waste based on its volume or weightage (Algorithm 4). The contract triggers once at least one company satisfies the conditions set in the smart contract such as weightage or volume, time of collection, and credits rewarded. Figure 7 presents an excerpt of the output of the function submitBid in the smart contract PaperWasteCollection. In this function, two companies submit their bids to collect recyclables offered by a client. The first company bids with 300 to collect 2 tons whereas the second company bids with 250 to recycle 1 ton. As the output of the smart contract shows, the first company’s bid is only accepted. Next, upon the collection of the waste, a 2-of-2 transaction is initiated that shifts the credits from the recycling company ledger to the client’s “Recycling Record” in the system. Specifically, the transferCredit function in the smart contract PaperWasteCollection is invoked, which transfers 300 credits from the recycling company to the client’s recycling record (Figure 8).

3.3 The case of unmatched recyclables

Addressing the challenge of unmatched recyclables is a fundamental aspect of the system’s operational framework. Instances may arise where items are procured and logged in the system but remain unrecycled by the purchasers. This issue can be attributed to a range of factors or circumstances impeding the successful recycling of those specific items. To address this issue, clients are afforded the opportunity to mitigate the impact of unmatched recyclables by reducing their purchase record. This can be accomplished through the process of recycling alternative items that qualify for redemption. By engaging in this practice, clients can reconcile the discrepancy between the purchased items and the actual recycling activities, ensuring the accuracy and integrity of the system’s records.

Figure 7. Remix IDE output for the submitBid function in the PaperWasteCollection smart contract in Algorithm 4.

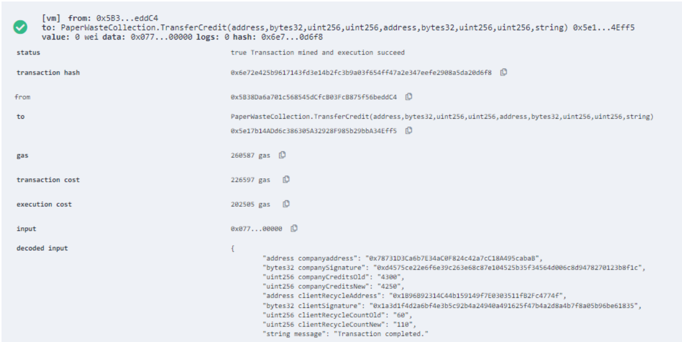

Figure 8. Remix IDE output for the transferCredit function in the PaperWasteCollection smart contract in Algorithm 4.

This approach allows for the effective management of unmatched recyclables, promoting transparency, accountability, and operational efficiency within the system. By providing clients with the means to rectify the situation through the recycling of suitable alternatives, the system fosters a streamlined and reliable recycling process while maintaining a comprehensive and accurate purchase record.

The system employs a value differentiation mechanism to account for the inherent variations among different items, considering their distinct characteristics, such as size and weight. This approach enables the system to establish equivalencies between items, exemplifying the ability to equate items of different quantities based on their defined value ratios. For instance, a one litre bottle is deemed equivalent to four 250 ml bottles, thus establishing a quantitative relationship that facilitates streamlined record-keeping and transactional operations within the system. This not only reduces the size of the ledger but also ensures that the system remains efficient in keeping track of the records.

Algorithm 5 illustrates the AdjustRecords smart contract that enables clients to reduce their purchasing records by recycling unmatched items. In the scenario where a client intends to enhance their recycling record by recycling a quantity of bottles that were not originally theirs, they can accomplish this by depositing the bottles at a designated recycling depot. Subsequently, the recycling depot updates the client's recycling record to accurately reflect the newly recycled items. At this point, both the recycling and the purchasing records contain unmatched items. To adjust the ledgers, the deleteItems function will undertake the following actions: (1) if the items in the recycling record are identical to those in the purchase record in terms of values, albeit with different identifiers, the system will delete the corresponding entries from both records (Figure 9); (2) if the items differ in value, the system will refer to the value of each item and equate them based on that, that is, one litre bottle versus a 250 ml bottle, the system may equate four items of the 250 ml size with one item of the one litre size and subsequently delete them from both records (Figure 10).

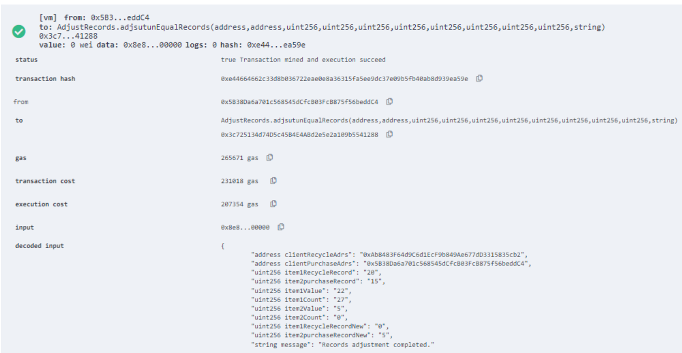

Figure 9. Remix IDE output for the transferCredit function in the AdjustRecords smart contract in Algorithm 5.

Figure 10. Remix IDE output for the transferCredit function in the AdjustRecords smart contract in Algorithm 5.

4. Conclusion

The utilisation of blockchain technology in recycling management has immense potential, and this proposed recycling system is a notable example of how this technology can be harnessed to address critical issues faced by the recycling industry. Our system ensures transparency and accountability in the recycling process. It is a multi-layered and multi-tiered modular blockchain-based recycling management system. The three distinctive layers of control, client, and IoT devices ensure that each stakeholder has a unique role and responsibility in the system. The use of smart contracts equipped with fully automated and secured management capabilities further enhances the efficiency of the system. Additionally, the implementation of the automated DPoS consensus mechanism ensures low overhead and less resource utilisation. The proposed system has the potential to revolutionise the recycling industry by providing a secure, transparent, and efficient platform for managing the recycling process. The proposed system exhibits diverse potential applications within the recycling industry. It can be effectively utilised by governments as an integral component of smart cities for waste disposal and recycling management. Furthermore, organisations can employ the system to monitor and enhance their waste management processes. Individuals can leverage its capabilities to facilitate efficient and environmentally conscious waste disposal practices. Additionally, recycling companies can utilise the system to streamline their operations and ensure effective tracking and management of recyclable materials.

Competing Interests:
None declared.

Ethical approval:
Not applicable.

Author's contribution:
MZ designed and coordinated this research and prepared the manuscript in entirety.

Funding:
None declared.

Acknowledgements:
None declared.

References

[1] S. Ponis, "Industrial symbiosis networks in Greece: Utilising the power of blockchain-based B2B marketplaces," *JBBA*, 2020. https://doi.org/10.31585/jbba-4-1-(4)2021.

[2] J. C. Borck and C. Coglianese, "Voluntary environmental programs: Assessing their effectiveness," *Annu. Rev. Environ. Resour.*, vol. 34, pp. 305–324, 2009.

[3] "Enhancing policy coherence for development: Workshop," Organisation for Economic Cooperation and Development, Tech. Rep., 2016, OECD Publishing.

[4] J. D. Ward and D. W. Gleiber, "Citizen response to mandatory recycling," *Public Product. Manag. Rev.*, vol. 16, no. 3, pp. 241–253, 1993.

[5] J. W. Everett and J. J. Peirce, "Curbside recycling in the U.S.A.: Convenience and mandatory participation," *Waste Manag. Res.*, vol. 11, no. 1, pp. 49–61, 1993.

[6] R. Pantcheva and G. Mengov, "Recycling rate in Europe: econometric modeling and dart clustering analysis," in *2022 Int. Conf. Automatics Inform.* (ICAI), 2022, pp. 179–182.

[7] A. Bongers and P. Casas, "The circular economy and the optimal recycling rate: A macroeconomic approach," *Ecol. Econ.*, vol. 199, p. 107504, 2022. [Online].

[8] S. Park, "Factors influencing the recycling rate under the volume-based waste fee system in South Korea," *Waste Manag.*, vol. 74, pp. 43–51, 2018.

[9] M. Paturi, S. Puvvada, B. S. Ponnuru, M. Simhadri, B. S. Egala, and A. K. Pradhan, "Smart solid waste management system using blockchain and IoT for smart cities," in *2021 IEEE Int. Symp. Smart Electron. Syst.* (iSES), 2021, pp. 456–459.

[10] E. Shanthini, V. Sangeetha, M. Jagadeeswari, B. Shivani, P. Selvapriya, K. Anindita, D. Divya Shree, and R. Suryanarayanan, "IoT based smart city garbage bin for waste management," in *2022 4th Int. Conf. Smart Syst. Inventive Technol.* (ICSSIT), 2022.

[11] X. Wang, D. Ma, and J. Hu, "Recycling model selection for electronic products considering platform power and blockchain empowerment," *Sustainability*, vol. 14, no. 10, p. 6136, 2022.

[12] P. K. Gopalakrishnan, J. Hall, and S. Behdad, "Cost analysis and optimization of blockchain-based solid waste management traceability system," *Waste Manag.*, vol. 120, pp. 594–607, 2021. [Online].

[13] S. Sahoo, A. Mukherjee, and R. Halder, "A unified blockchain-based platform for global e-waste management," *Int. J. Web Inf. Syst.*, vol. 17, no. 5, pp. 449–479, 2021.

[14] A. U. R. Khan and R. W. Ahmad, "A blockchain-based IoT-enabled e-waste tracking and tracing system for smart cities," *IEEE Access*, vol. 10, pp. 86256–86269, 2022.

[15] M. K. Nallapaneni and S. S. Chopra, "Blockchain-based online information sharing platform for improving the resilience of industrial symbiosis-based multi energy systems," in *Actionable Science for Urban Sustainability 2020 (AScUS-2020): AScUS Unconference*, 2020.

[16] T. Ding, G. Yan, Z. Zhou, and Y. Lei, "Research on product life cycle data traceability based on multiblockchain," in *2021 3rd Int. Symp. Robot, Intell. Manuf. Technol.* (ISRIMT), 2021.

[17] C. Wankm¨uller, J. Pulsfort, M. Kunovjanek, R. Polt, S. Craß, and G. Reiner, "Blockchain-based tokenization and its impact on plastic bottle supply chains," *Int. J. Prod. Econ.*, vol. 257, p. 108776, 2023.

[18] S. Pandey, V. Chouhan, D. Verma, S. Rajrah, F. Alenezi, R. Saini, and K. Santosh, "Do-it-yourself recommender system: Reusing and recycling with blockchain and deep learning," *IEEE Access*, vol. 10, pp. 90056–90067, 2022.

[19] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System.* Bitcoin.org, 2008.

[20] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. O'Reilly Media, Inc., 2014.

[21] J. Sun, L. Ren, S. Wang, and X. Yao, "A blockchain-based framework for electronic medical records sharing with fine-grained access control," *PLOS ONE*, vol. 15, no. 10, pp. 1–23, 2020.

[22] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE Int. Congr. Big Data* (BigData Congress), 2017, pp. 557–564.

[23] J. Shah and S. Parveen, "Understanding the blockchain technology beyond bitcoin," In *Advances in Industrial and Production Engineering*: Select Proceedings of FLAME 2020, pp. 499–516, Springer.

[24] V. Ram, K. C. Kishore, and S. N. Kalidindi, "Environmental benefits of construction and demolition debris recycling: Evidence from an Indian case study using life cycle assessment,"*J. Clean. Prod.*, vol. 255, p. 120258, 2020. https://www.sciencedirect.com/science/article/pii/S095965262030305X.

Appendix

Algorithm 1 Adding purchase entry to client's record

```
contract VendingMachine.
Declare:
struct PurchaseTransactions(clientPurchaseAdrs, vendingMachineAdrs, itemId, clientSignature, vendMachineSignature, purchaseCountOld, purchaseCountNew, message).
mapping purchaseTransaction of type PurchaseTransactions.
struct PurchasedItems(clientPurchaseAdrs).
mapping purchasedItem type of PurchasedItems.
event NewpurchaseTransaction(clientPurchaseAdrs, vendingMachineAdrs, itemId, clientSignature, vendMachineSignature, purchaseCountOld, purchaseCountNew, message).
verify: signature of client.
if 1 then
function processItem(itemId,items[])
{
N ← length(items[])
    require (!(N¡ limit), "Limit reached, please recycle")
purchasedItem[clientPurchaseAdrs].push(itemId)
}
    emit NewpurchaseTransaction
end if
```

Algorithm 2 Adding recycle entry to client's record

```
contract RecyclingDepot.
Declare:
struct Transactions(clientRecycleAdrs, clientPurchaseAdrs, itemId, clientSignature, depotSignature, recycleCountOld, recycleCountNew, purchaseCountOld, purchaseCountNew, message).
mapping Transaction of type Transactions.
struct Items (itemId; owner; itemName; itemValue;)
mapping recycledItem type of Items.
event NewTransactions(clientRecycleAdrs, clientPurchaseAdrs, itemId, clientSignature, depotSignature, recycleCountOld, recycleCountNew, purchaseCountOld, purchaseCountNew, message).
verify: signature of client.
if 1 then
   function processItem(itemId, purchasedItem[])
{
   N ← length(purchasedItem[])
   for i ← 1 to N do
      if (purchasedItem[i]==itemId) then purchasedItem[i]=purchasedItem[length(purchasedItem[])-1];
          items.pop();
          emit NewTransactions
      else recycledItem[clientRecycleAdrs].push(itemId)
          emit NewTransactions
      end if
   end for
}
end if
```


Algorithm 3 Recyclables collection contract

```
contract RecyclingContract.
Declare:
collectionFee; isAgreementAccepated; isTransactionCompleted;
struct Transactions(isAgreementAccepated, collectionFee; numberOfItems; restaurantPurchaseAdrs; companyRecycleAdrs; restau-
rantSignature; companySignature; rstntPurchaseCountOld; rstntPurchaseCountNew; cmpnPurchaseCountOld; cmpnPurchaseCount-
New; isTransactionCompleted).
mapping Transaction of type Transactions.
struct Items (itemId; owner; itemName; itemValue;)
mapping recycledItem type of Items.
mapping purchasedItem type of Items.
event AgreementAccepted(companyRecycleAdrs)
function acceptAgreement() {
require (msg.sender==!restaurantPurchaseAdrs)
isAgreementAccepted=true;
companyRecycleAdrs=payable(msg.sender)
emit AgreementAccepted(companyRecycleAdrs); }
function executeTransaction() payable {
require (msg.sendr==restaurantPurchaseAdrs)
require (isAgreementCompleted==true)
require (isTransactionCompleted==false)
require (msg.value==collectionFee)
N ← length(purchasedItems[restaurantPurchaseAdrs])
for i ← 1 to N do
    purchaseValue = purchasedItems[restaurantPurchaseAdrs][i].itemValue + purchaseValue;
end for
i=0;
while (purchaseValue>0) do
    purchasedItems[restaurantPurchaseAdrs][i]=
purchasedItems[restaurantPurchaseAdrs][length(purchasedItems[restaurantPurchaseAdrs])-1];
    recycledItems[companyRecycleAdrs][length(recycledItems[companyRecycleAdrs])+1]=
purchasedItems[restaurantPurchaseAdrs][i];
    purchasedItems.pop();
    purchaseValue =purchaseValue- purchasedItems[restaurantPurchaseAdrs][i].itemValue;
    i++;
end while
isTransactionCompleted← true;
companyRecycleAdrs.transfer(msg.value);
emit TransactionCompleted }
```

Algorithm 4 Untraceble recyclables

```
contract PaperWasteCollection.
Declare: companyAdrs; recyclceCompanyAdrs; paperWeight; paperVolume; credits; contractFulfilled;
constructor( _weight, _volume, _credits) {
paperWeight = _weight; paperVolume = _volume; collectionTime = Time; contractFulfilled = false; credits=_credits;
emit ContractCreated( paperWeight, paperVolume, collectionTime, credits); }
function submitBid(companyAdrs, companyBid, companyVol) public {
require (msg.sender != companyAdrs);
require(!contractFulfilled);
emit BidSubmitted(msg.sender, bid);
if (bid ¿= credits) then
    recyclceCompanyAdrs = msg.sender;
    contractFulfilled = true;
    emit ContractFulfilled(companyAdrs, recyclceCompanyAdrs, credits);
end if} }
event ContractCreated(companyAdrs, paperWeight, paperVolume, collectionTime, credits);
event BidSubmitted(recyclceCompanyAdrs, bid);
event ContractFulfilled(companyAdrs, address recyclceCompanyAdrs, credits);
function transferCredits()
require(contractFulfilled);
emit transferCredit(recyclceCompanyAdrs, companyAdrs, creditsToTransfer);
```

Algorithm 5 Unmatched records adjustment

```
contract AdjustRecords
Declare:
struct RecycleTransactions(clientRecycleAdrs; itemValue; clientSig; depotSig).
mapping recycleTransaction of type RecycleTransactions.
struct PurchaseTransactions(clientPurchaseAdrs; itemValue; clientSig; depotSig).
mapping purchaseTransaction of type PurchaseTransactions.
struct Items (itemId; owner; itemName; itemValue;)
 mapping  recycledItem type of Items.
 mapping  purchasedItem type of Items.
event New recycleTransactions(transactionId, clientRecycleAdrs, itemId, clientSig, depotSig).
event New purchaseTransactions(transactionId, clientPurchaseAdrs, itemId, clientSig, depotSig).
function deleteItems(recycledItems[], purchasedItems[] )
N ← length(recycledItems[])
for i ← 1 to N do
    recycleValue = recycledItems[i].itemValue + recycleValue;
end for
m ← length(purchasedItems[])
for j ← 1 to M do
    purchaseValue = purchasedItems[j].itemValue + purchaseValue;
end for
if (purchasedalue¿recycleValue) then
    j=0;
    while (purchaseValue¿recycleValue) do
        purchasedItems[j]=purchasedItems[length(purchasedItems[])-1];
        purchasedItems.pop();
        purchaseValue =purchaseValue- purchasedItems[j].itemValue;
        j++;
    end while
    for i ← 1 to N do
        recycledItems[i].pop();
    end for
else
    if (purchaseValue¡recycleValue) then
        i=0;
        while (purchaseValue¡recycleValue) do
            recycledItems[i]=recycledItems[length(recycledItems[])-1];
            recycledItems.pop();
            recycleValue ==recycleValue- recycledItems[i].itemValue;
            i++
        end while
        for j ← 1 to M do
            recycledItems[j].pop();
        end for
    else
        i=0;
        while (purchaseValue=recycleValue) do
            for j ← 1 to M do
                recycledItems[j].pop();
            end for
            for i ← 1 to N do
                recycledItems[msg.sender][i].pop();
            end for
        end while
    end if
end if
emit New purchaseTransactions
emit New recycleTransactions
```