mathematics

MDPI

Article

A Dynamic Trading Approach Based on Walrasian Equilibrium in a Blockchain-Based NFT Framework for Sustainable Waste Management

Ch Sree Kumar [1], Aayushman Bhaba Padhy [2], Akhilendra Pratap Singh [1] and K. Hemant Kumar Reddy [3,*]

[1] Department of Computer Science and Engineering, National Institute of Technology Meghalaya, Shillong 793003, India; p21cs013@nitm.ac.in (C.S.K.); akhilendra.singh@nitm.ac.in (A.P.S.)
[2] Department of Computer Science and Engineering, NIST Institute of Science and Technology (Autonomous), Berhampur 761008, India; apb121@nist.edu
[3] Department of Computer Science and Engineering, VIT-AP University, Amaravati 522237, India
* Correspondence: hemanth.reddy@vitap.ac.in

Abstract: It is becoming harder to manage the growing amounts of waste generated daily at an increasing rate. These problems require an efficient solution that guarantees effectiveness and transparency and maintains trust within the community. To improve the process of traditional waste management, we proposed a unique solution, "GREENLINK", which uses a combination of blockchain technology with the concept of zero-knowledge proofs (ZKPs), non-fungible tokens (NFTs), and Walrasian equilibrium. Zero-knowledge proofs (cryptographic protocols) are used to verify organizations and prove compliance (e.g., certification, recycling capacity) without disclosing sensitive information. Through an iterative bidding process, the proposed framework employs Walrasian equilibrium, a technique to balance supply and demand, guaranteeing equitable pricing and effective resource distribution among participants. The transactions and waste management activities are securely recorded on an immutable ledger, ensuring accountability, traceability, and transparency. The performance of the proposed model is evaluated. Parameters like average latency, TPS, and memory consumption are calculated using Hyperledger Caliper (a blockchain performance benchmark framework).

Keywords: non-fungible tokens (NFTs); blockchain; zero-knowledge proofs (ZKPs); waste management; decentralized system; Walrasian equilibrium; dynamic bidding; metamask; blockchain; Hyperledger Caliper

MSC: 91A35

check for updates

Academic Editor: Óscar Valero Sierra

Received: 15 December 2024
Revised: 19 January 2025
Accepted: 23 January 2025
Published: 5 February 2025

Citation: Kumar, C.S.; Padhy, A.B.; Singh, A.P.; Reddy, K.H.K. A Dynamic Trading Approach Based on Walrasian Equilibrium in a Blockchain-Based NFT Framework for Sustainable Waste Management. *Mathematics* **2025**, *13*, 521. https://doi.org/10.3390/math13030521

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Every day, huge amounts of waste are generated by residential, commercial, and medical sources. Traditionally, the recycling process of this waste material is handled by government organizations, like municipalities, or private organizations. These entities use their resources, such as garbage trucks and employees, to collect waste from various locations, including houses, apartments, medical institutions, industries, etc. However, this process involves a leak of sensitive/personal information of users, such as names, addresses, payment information, and phone numbers. These data are accessible to employees, posing risks such as planned robberies and misuse of waste materials. Loss of materials, unscientific treatment, improper collection of waste, and ethical problems are just a few of the problems that these organizations face. In this case, the user is basically not aware of the

whole process of waste management, does not have control over their data, and is unaware of the authenticity of the organization in charge. Such things erode trust between companies and users. Several organizations provide end-to-end waste management services like collection, segregation, transportation, recycling, composting, incineration, landfilling, hazardous waste treatment, regulatory compliance, and record keeping. Here, all the data generated during these processes are stored centrally. This has drawbacks, such as a single point of failure, scalability limitations, performance bottlenecks, data privacy concerns, etc. Moreover, the verification of waste recycling companies is also an issue. Organizational backdoor entries pose a major risk of cheating customers.

The advantages of using blockchain in waste management are transparency, traceability, efficiency, environmental impact, accountability, etc. The pros of the non-fungible token-based approach to waste management are traceability and transparency (including acting as a digital passport for waste, real-time tracking, preventing illegal dumping), incentivizing recycling and sustainable practices (including rewarding eco-conscious behavior, gamification of recycling, funding sustainable initiatives), ensuring ethical practice, fair trade (fair compensation for recyclers, verifying the origin of recycled materials) and environmental impact tracking, improved public engagement and trust, community-driven initiatives, reduced fraud and corruption, and optimized resource allocation.

To address these challenges, we propose Greenlink—a blockchain-based NFT waste management framework that streamlines waste recycling by connecting verified companies and organizations. The NFT tokens are accessible to all authorized stakeholders, and verified organizations are available to carry out the recycling process efficiently and securely. Dynamic trading refers to a real-time or near-real-time process in which an automated bidding system determines the price for a good, service, or advertisement slot. It is commonly used in online advertising, e-commerce, and auction platforms. Dynamic bidding optimizes pricing to reflect real-time market conditions, increases efficiency and ROI, and enhances competitiveness in the marketplace. The Walrasian equilibrium proposed in our model is a state in a market where supply equals demand, and every participant (buyer or seller) maximizes their utility at given prices. This concept is instrumental in dynamic trading systems for the efficient allocation of resources, real-time price discovery, maximizing platform revenue, stability and predictability, and multi-item auctions. By leveraging Walrasian equilibrium principles, dynamic trading systems enhance fairness, efficiency, and market effectiveness. The Walrasian equilibrium approach aims to find a price vector that balances supply and demand in a market. In this waste bidding model, prices are iteratively adjusted based on excess supply or demand until equilibrium is reached. The verification process is carried out using zero-knowledge proofs, which allows companies to be verified without revealing their private information to the administrator who sets the criteria. Zero-knowledge proofs are an efficient technique to minimize the information transferred between the prover and the verifier in a cryptographic protocol. The prover can prove through a cryptographic commitment scheme to another party, termed the verifier, without revealing how it is proven. Zero-knowledge proofs, or ZKPs, are a technique for one party to cryptographically prove to another that they know something about a piece of information without disclosing the underlying information. A ZKP in the context of blockchain networks may only disclose on-chain that a piece of secret information is legitimate and is known by the prover with high certainty. It operates when the verifier urges the prover to conduct activities that can only be completed correctly if the prover is aware of the fundamental information. If the one who proves is guessing about the outcome of these acts, the verifier's test will almost certainly prove them incorrect with a degree of probability. Zero-knowledge proofs (ZKPs) are defined by key characteristics that make them a powerful, secure, and private verification tool. Accuracy ensures that a

reliable prover can convince an honest verifier that they know the correct input if a claim is true. Stability guarantees that no unethical prover can deceive the verifier into believing they possess the correct information if the claim is false. Zero expertise means that if the claim is true, the verifier learns nothing more from the prover except that the claim is valid. These features make ZKPs a robust method for proving knowledge without revealing additional information. In interactive ZKPs, the prover and verifier communicate back and forth, allowing the verifier to validate the claim, while in non-interactive ZKPs, the prover generates a proof using an algorithm, which is then validated by another algorithm.

Using lightweight protocols like Mina and Polygon zkEVM greatly benefits GREENLINK, making the platform energy-efficient. The Mina protocol and Polygon zkEVM use lightweight zk-SNAKS and layer-2 blockchain scaling solutions. The design of the Mina protocol architecture is unique. It has proven to use less memory and CPU because the blockchain size is fixed at 22kb, regardless of the number of transactions. This is made possible by storing images instead of all transactions. This also eliminates the need to install huge storage systems for running the blockchains' nodes and utilizes less power than POW or POS blockchains like Bitcoin. On the other hand, Polygon zkEVM uses the concept of zk-rollups, which follows the approach of winding or rolling multiple transactions into a single form and performing verifications using ZKPs. This approach reduces the computational power and energy costs associated with the transaction process and verification of the data. By performing the transaction computation operation off-chain, which makes the use of smart contracts less and using efficient cryptographic methods, Polygon zkEVM provides high scalability and lower energy consumption options. The performance benchmarks conducted using Hyperledger Caliper ensure that the framework operates efficiently, with low latency, high TPS (transactions per second), and reduced memory consumption translating to less energy use per transaction. By optimizing resource utilization, the framework minimizes its environmental impact during operation.

The proposed framework, Greenlink, includes scalable and efficient features to alleviate the worries about any bottlenecks the system may experience due to its dependence on smart contracts and zero-knowledge proofs using the following features. (A) Use of scalable blockchain infrastructure: For on-chain validation, the proposed framework, Greenlink, uses Polygon zkEVM, which collects data into a single forum and compresses many transactions into a single batch via zk-rollups. This preserves the system's integrity and security while significantly lowering the strain on the main blockchain. Thanks to the use of zk-rollups, Greenlink can handle thousands of transactions per second (TPS), which makes it capable of handling large user traffic and performing operations even during heavy workloads. (B) The Mina protocol's lightweight architecture: The 22 KB size of the Mina blockchain makes it highly efficient in handling the application's memory usage compared to any other blockchain. The architecture guarantees that the system maintains effectiveness even with increased user demand. (C) Optimization of smart contracts: To reduce computing complexity and gas costs, the proposed framework's smart contracts are designed to only use essential information required for a successful waste management process. The framework minimizes the risk of network congestion and maximizes the execution time by using the latest technologies. (D) Event-driven execution: Due to their event-driven nature, smart contracts save needless computing costs by only operating when certain criteria are satisfied. (E) Enhancements in zero-knowledge proofs (ZKPs): The proposed framework, Greenlink, makes sure that—even with high user loads—the ZKP process does not create a bottleneck by optimizing the timings for ZKP production and verification. Our approach aggregates and verifies ZKPs for several transactions together, significantly lowering the computational cost of the verification process. (F) Layer-2 integration: By enabling off-chain processing for most processes, layer-2 solutions, such as

Polygon zkEVM, lessen dependency on the layer-1 blockchain and prevent congestion. (G) Performance benchmarking: Using Hyperledger Caliper, Greenlink tests metrics, including average latency, TPS, and memory usage, under simulated high loads to assess its performance. This guarantees that the system is highly scalable and offers information about any bottlenecks for future enhancements.

The organization of the paper is as follows: Section 1 provides the introduction; Section 2 includes a review of the literature; Section 3 discusses blockchain-enabled NFT frameworks with zero-Knowledge proof (ZKP)-based verification for waste management; Section 4 discusses their implementation and the results; and Section 5 provides the conclusion.

2. Literature Review

This section reviews the latest work in different areas of non-fungible tokens and works, including OpenZeppelin in blockchain technology, zero-knowledge proofs (ZKPs), and the Mina protocol, which relates to the topic of waste management in some way, either directly or indirectly. Researchers worldwide have used these concepts in various areas such as waste management, supply chain management, healthcare, drug traceability, etc. This section briefly describes the research, diving into broader areas such as trading approach, traceability systems, NFT-based artworks, security, and auction methodology. The role of OpenZeppelin contracts is to define the NFTs' buy, sell, and minting functionalities.

ZKP and blockchain technology have the potential to completely transform the waste management industry and the way the entire process is managed. In [1], Khiem et al. proposed an innovative solution for medical waste management, integrating blockchain, smart contracts, and NFTs. In [2,3], Bulkowska et al. and Jiang et al. discussed and implemented blockchain in waste management. In [4], Tian et al. discussed post-carbon footprint analysis in the art market, and in [5], Mielcarek et al. showcased water nutrient management. In [6], Chiacchio et al. discussed NFTs in the pharmaceutical sector, while in [7], Chiquito et al. discussed waste management in aquaculture, and in [8], Rayes et al. proposed a plastic waste management model. Triet et al. proposed a framework in [9] that combined blockchain with circular economy principles to improve construction waste management. In [10], Ahmed et al. developed a framework to examine Blockchain's potential to enhance waste management in smart cities and discussed security. In [11], Yuan et al. proposed an IoT-based solid waste management system for developing countries, utilizing blockchain-enabled VANETS, UHF technology, and geo-fencing for real-time tracking and communication. Steenmans et al., in [12], examined blockchain's role in plastic waste management, identifying key areas like payment, recycling rewards, waste tracking, and smart contracts.

Blockchain technology offers promising benefits to supply chain management by enhancing transparency, security, and efficiency through its decentralization feature and immutable ledger technology, which eliminates reliance on third-party dependencies. In [13], Farsi et al. examined blockchain-based supply chain systems, highlighting their benefits and key requirements while identifying potential cyber threats and security challenges. Soori et al., in [14], explored how integrating blockchain with IIoT in Industry 4.0 enhances sustainable supply chain management, offering real-time tracking and reduced intermediaries. Aslam et al. proposed a framework for evaluating whether a complex supply chain should adopt blockchain technology in [15]. They empirically analyzed the supply chain practices of Pakistan's oil industry and their impact on operational performance. In [16,17], Dietrich et al. and Duan et al. reviewed the most recent publications on blockchain in supply chain management, classifying them by complexity. They found that most blockchain projects focus on simple supply chains, with no examples yet addressing transparency and

audibility in complex manufacturing supply chains. In [18,19], Dudczyk et al. and Marin et al. discussed blockchain platforms in global supply chain management and distributed technology, their industry applications, and existing solutions.

The adoption of blockchain technology in emerging trading and auctions seems promising and can help enhance efficiency, reduce scams, and form trust between trading parties. Mariia Rodinko et al., in [20], proposed upgrading existing cryptocurrencies to new tokens without relying on external information sources to avoid KYC, ensuring a predicted supply and fair pricing based on decentralized auction simulations. In [21], Cui et al. proposed a novel competition padding auction mechanism for peer-to-peer energy trading among renewable prosumers, addressing the budget deficit issue. In [22], Kadadha et al. proposed AB Crowd, a fully decentralized crowdsourcing framework using the Ethereum blockchain and the repeated single-minded bidder auction mechanism. In [23], Agarwal et al. explored a decentralized blockchain-based double auction mechanism for energy trading in a smart microgrid, allowing prosumers to sell surplus energy to consumers. In [24], Liu et al. addressed computation offloading in mobile blockchain networks by proposing a smart contract-based double auction mechanism.

In [25], Yang et al. proposed a blockchain-based digital identity management system using smart contracts, ZKPs, and a BZDIMS prototype for enhanced privacy. Wali et al., in [26], discussed secure vehicle data management using blockchain and ZKPs. In [27], Soewito et al. study combined AES and ZKPs for authentication, improving data transmission security and reducing authentication processing time. In [28], Diro et al. discussed identity sharing with ZKPs and blockchain for secure identity sharing, addressing vulnerabilities. In [29], Kuznetsov et al. proposed an innovative aggregation scheme for ZKPs within Merkle trees to enhance data verification efficiency in blockchain systems. Experimental results show significant reductions in proof size and computational requirements, offering a scalable and secure solution for various applications. In [30], Xiao Xu et al. explored using blockchain-powered zero-knowledge proofs to create a privacy-preserving disability management system that enhances inclusivity.

The current paper proposes a model that integrates technologies like zero-knowledge proofs (ZKPs), blockchain, and the Mina protocol (a lightweight blockchain that only stores the image of a blockchain) to efficiently handle waste management using a novel dynamic or repeated bidding using Walrasian equilibrium for iterative price adjustments to sell waste and allot private organizations that are verified and genuine. After the ZKP-based verification method, private organizations can participate in the auction process to purchase waste from individuals or organizations in the NFT marketplace. Digital assets are stored via decentralized storage, i.e., NFT.storage, which is is a specialized service that provides decentralized functionalities for non-fungible tokens. We have used its API endpoint to successfully store the tokens in its storage. The Coinbase Wallet SDK enables users to connect to Greenlink through the Coinbase Wallet. The Web3Modal library makes connecting decentralized applications to different cryptocurrency wallets simple. The benchmarking tool—Hyperledger Caliper is used to calculate the throughput and efficiency of the entire platform (Table 1).

Table 1. Comparison of the proposed system with other existing systems.

References	Framework/Model Used	Performance Evaluation	Objective
[1]	Hyperledger fabric, NFT framework	Transaction throughput: 200 TPS	Address waste management using blockchain in Vietnam.
[9]	Ethereum-based smart contracts	Reduced waste mis management by 30%	Streamlining medical waste management in healthcare.
[10]	Multi-chain architecture for smart cities	Cost efficiency improved by 20%	Blockchain for smart city waste management.
[11]	Decentralized information management system	Operational efficiency +25%	Sustainable construction and demolition waste management.
[12]	Public blockchain with smart contract governance	Recycling accuracy +15%	Plastic waste management governance using blockchain.
[13]	Blockchain-based supply chain model	Improved security by 40%	Securing supply chains with blockchain.
[14]	Industrial IoT model integrated with blockchain	Reduced energy consumption in SC	Sustainable supply chain management in Industry 4.0.
[15]	Private blockchain adoption model	Reduction in delays by 20%	Blockchain in oil industry supply chains.
[17]	Blockchain maturity model	Increased ROI by 35%	Application of blockchain in SCM.
[18]	Consortium blockchain	Lowered logistics cost by 15%	Survey of blockchain applications in global SCM.
[20]	Proof-of-Burn consensus	System efficiency: 92%	Decentralized cryptocurrency upgrade mechanism.
[21]	Consortium and double auction	P2P energy trading success rate: 95%	Blockchain for energy trading with double auction.
[22]	Auction mechanism integrated with blockchain	Computational efficiency +20%	Crowdsourcing auction mechanism using blockchain.
[23]	Double auction mechanism	Energy savings: +15%	Smart microgrid energy trading using blockchain.
[24]	Smart contract-based computation offloading	Reduced latency by 20%	Mobile blockchain computation offloading.
[25]	Zero-knowledge-proof-based identity scheme	Improved privacy by 30%	Blockchain identity management using ZKP.
[26]	ZKAV framework	Enhanced AV performance	Blockchain for autonomous vehicles.
[27]	Modified ZKP algorithm	Faster authentication by 15%	IoT security system with ZKP based verification.

3. Blockchain-Enabled NFT Framework

Waste management is a process and a set of actions that are required to manage waste, starting from its generation to its final disposal. Effective waste management helps reduce waste generation, increase recycling and reuse, minimize environmental impacts (e.g., pollution, climate change), protect public health and safety, and conserve natural resources. NFTs are paving the way to effectively help in waste management in the following ways: tracking and verification, waste ownership and responsibility, incentivizing recycling, supply chain optimization, carbon credit verification, digital twins for waste management, education and awareness, and decentralized waste management. The NFT

blockchain-based waste management trading framework uses decentralized technology to improve waste buying, selling, and ownership. This framework ensures the organization's verification using the zero-knowledge proof (ZKP) method. Provides secure, transparent transactions, reduces the risk of fraud, and enables seamless ownership transfers using smart contracts. The immutable nature of blockchain records each transaction, providing verifiable provenance for NFTs. The framework also provides an auction-based mechanism for purchasing goods by verified organizations, and financial transactions are stored securely on the blockchain.

3.1. Proposed Mathematical Model

To model a bidding process in which organizations and users purchase waste from companies, we can create a mathematical model involving the following components (Table 2):

Table 2. Notations and descriptions.

Notation	Description
$i \in \{1, \ldots, N\}$	Index for waste-selling companies.
$j \in \{1, \ldots, M\}$	Index for waste-buying organizations/users.
Q_i	Quantity of waste available from company i (in units).
Q_{ij}	Quantity of waste allocated by company i to organization j (decision variable).
P_{ij}	Bid price per unit of waste submitted by organization j for company i.
C_i	Reserve price per unit set by company i (minimum acceptable price).
B_j	Budget of organization/user j.
$U_j(Q_{ij})$	Utility or value derived by organization j for acquiring Q_{ij} units of waste.
$R_i(Q_i)$	Revenue function for company i.

Assumption 1. *1. The allocation of waste quantity cannot exceed the available supply or budgetary limits.*

2. Each organization maximizes its utility while each company maximizes its revenue.

3. Prices are determined based on bidding values and reserve prices.

Objective Functions:

1. Company Objective: Maximize total revenue:

$$R_i = \sum_{j=1}^{M} P_{ij} Q_{ij}, \quad \text{subject to} \quad P_{ij} \geq C_i, \quad \forall j \tag{1}$$

2. Organization/User Objective: Maximize total utility:

$$U_j = \sum_{i=1}^{N} U_j(Q_{ij}) - \sum_{i=1}^{N} P_{ij} Q_{ij}, \quad \text{subject to} \quad \sum_{i=1}^{N} P_{ij} Q_{ij} \leq B_j, \quad \forall j \tag{2}$$

The constraints of the mathematical model are as follows.

1. Quantity Constraint for Companies:

$$\sum_{j=1}^{M} Q_{ij} \leq Q_i, \quad \forall i \tag{3}$$

2. Budget Constraint for Organizations/Users:

$$\sum_{i=1}^{N} P_{ij} Q_{ij} \leq B_j, \quad \forall j \tag{4}$$

3. Non-Negativity:

$$Q_{ij} \geq 0, \quad \forall i, j \tag{5}$$

4. Bid Price Constraint:

$$P_{ij} \geq C_i, \quad \forall i, j \tag{6}$$

Assuming that the problem involves dynamic or repeated bidding, iteratively adjust prices P_{ij} and allocations Q_{ij} to clear the market (i.e., supply equals demand). The Walrasian equilibrium approach seeks a price vector that balances supply and demand in a market. We iteratively adjust prices based on excess demand or supply for this waste bidding model until equilibrium is achieved.

3.2. Mathematical Modelling of the Dynamic Bidding Using Walrasian Equilibrium

1. Initialization: Set an initial price vector

$$P^0 = [P_{ij}^0] \tag{7}$$

where P_{ij}^0 is the price for each pair for company-buyer (i, j).
Ensure

$$P_{ij}^0 \geq C_i \tag{8}$$

for all i, j.

2. Excess Demand Calculation: At each iteration t, compute the excess demand for each company i based on the current price P^t:

$$Z_i^t = \sum_{j=1}^{M} D_{ij}(P_{ij}^t) - Q_i \tag{9}$$

where $D_{ij}(P_{ij}^t)$ is the demand from buyer j for waste from company i at price P_{ij}^t, computed as follows:

$$D_{ij}(P_{ij}^t) = \arg\max_{Q_{ij}} \left(U_j(Q_{ij}) - P_{ij}^t Q_{ij} \right) \tag{10}$$

subject to $Q_{ij} \leq \frac{B_j}{P_{ij}^t}$.

3. Price Adjustment: Update prices iteratively based on excess demand Z_i^t :

$$P_{ij}^{t+1} = P_{ij}^t + \alpha \cdot Z_i^t \tag{11}$$

where $\alpha > 0$ is a step size controlling the rate of price adjustment.

4. Stopping Criterion: Continue iterations until

$$|Z_i^t| \leq \epsilon, \quad \forall i \tag{12}$$

where ϵ is a small threshold indicating equilibrium (i.e., supply equals demand).

5. Final allocation: Once equilibrium prices P_{ij}^* are found, allocate waste quantities:

$$Q_{ij}^* = D_{ij}(P_{ij}^*) \tag{13}$$

At equilibrium: Prices: equilibrium prices P_{ij}^* and Quantities Q_{ij}^* satisfy

$$\sum_j Q_{ij}^* = Q_i \tag{14}$$

(Supply equals demand for each company).

3.3. Proposed Framework/Approach—Greenlink

We have proposed a blockchain-based approach for waste management called Greenlink. Greenlink exclusively facilitates waste recycling among verified companies and organizations. Using the zero-knowledge proof method; we ensure the authenticity of the participants. Recyclable waste is auctioned off to interested organizations. The highest bidder purchases the waste, which is then sold through our marketplace. The proposed framework uses the Mina protocol, a lightweight blockchain, as it stores only the image of the blockchain. The architecture of the proposed model is shown in Figure 1.

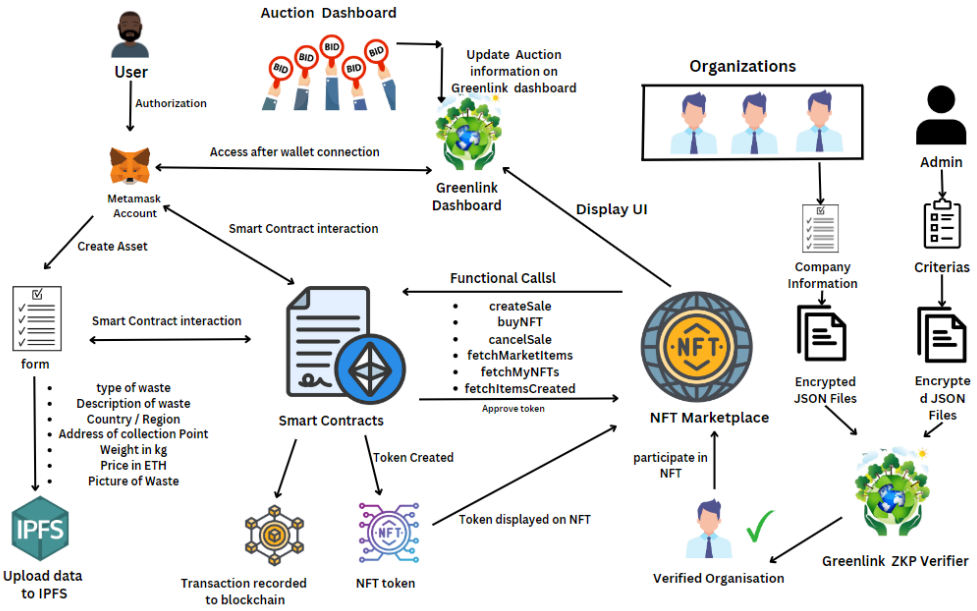


Figure 1. Architecture of the proposed model—Greenlink.

The functionality of the proposed framework—Greenlink—is divided into two parts. The primary functionality is to help people recycle their waste safely and securely through the use of the decentralized marketplace, where users can upload detailed information about waste, such as category, weight, address, etc. The other functionality is to let only valid waste management organizations participate in the decentralized marketplace. Organizations are verified using the zero-knowledge proof (ZKP) method. The proposed marketplace facilitates the sale of recyclable waste with auctions, ensuring competitive pricing and secure transactions for verified participants. The NFT tokens are accessible to all authorized stakeholders, and verified organizations will be available to carry out the recycling process efficiently and securely. Polygon zkEVM provides a scalable environment in which the EVM compatibility and zk-rollups are grouped for the efficient running of smart contracts with Ethereum. A cryptographic primitive known as zero-knowledge proof

(ZKP) is used by the Polygon zkEVM for validating the state transitions. It scales up Ethereum. Transaction confirmations are quicker, and the fees are lower when compared to Ethereum MainNet, which helps design complex blockchain applications.

3.4. Proposed Framework's Algorithms

In our currently developed model, NFT token trading operations were implemented using smart contracts. The following algorithms were developed and used in our framework.

Algorithm 1 shows the minting operation of waste materials and its listing in NFT marketplace.

Algorithm 2 shows the creation of a market item.

Algorithm 3 reflects the transfer of ownership of items and funds between parties.

Algorithm 4 depicts the fetching of unsold market items.

Algorithm 5 shows the fetching operation of items the user has purchased.

Algorithm 6 shows the process of zero-knowledge proof.

Algorithm 7 shows dynamic or repetitive bidding using the Walrasian equilibrium approach.

Algorithm 1 Minting waste and listing in marketplace

```
function CREATETOKEN(tokenURI (String), price (Integer)) Integer
    Increment the token ID counter:
        _tokenIds.increment()
    Get the current token ID and assign it to newTokenId:
        newTokenId = _tokenIds.current()
    Mint a new token to the sender with the new token ID:
        _mint(msg.sender, newTokenId)
    Set the token URI for the new token ID to the provided tokenURI:
        _setTokenURI(newTokenId, tokenURI)
    Create a market item with the new token ID and the provided price:
        createMarketItem(newTokenId, price)
    Return the new token ID:
        return newTokenId
end function
```

Algorithm 2 Creating a market item

```
function CREATEMARKETITEM(tokenId: Integer, price: Integer)
    if price < 0 then
        Raise error "Price must be at least 0 wei"
    end if
    newMarketItem ← Create MarketItem with:
    tokenId = tokenId
    seller = sender's address
    owner = contract's address
    price = price
    sold = false
    idToMarketItem[tokenId] ← newMarketItem
end function
```

Algorithm 3 Transferring ownership of the item, as well as funds between parties

function CREATEMARKETSALE(tokenId)
    price ← idToMarketItem[tokenId].price
    seller ← idToMarketItem[tokenId].seller
    if msg.value ≠ price **then**
        Raise error "Please submit the asking price to complete the purchase."
    end if
    idToMarketItem[tokenId].owner ← sender's address
    idToMarketItem[tokenId].recycled ← true
    idToMarketItem[tokenId].seller ← null address
    Increment the itemsRecycled counter
    Transfer the token from contract's address to sender's address
    Transfer the payment to the seller
end function

Algorithm 4 Fetching all unsold market items

function FETCHMARKETITEMS
    return array of MarketItem
    $itemCount \leftarrow$ current value of _tokenIds
    $unsoldItemCount \leftarrow$ current value of _tokenIds - current value of _itemsRecycled
    $currentIndex \leftarrow 0$
    Create a new array *items* of type MarketItem with length $unsoldItemCount$
    for i **from** 0 **to** $itemCount - 1$ **do**
        if owner of $idToMarketItem[i+1]$ is equal to contract's address **then**
            $currentId \leftarrow i+1$
            $currentItem \leftarrow idToMarketItem[currentId]$
            $items[currentIndex] \leftarrow currentItem$
            $currentIndex \leftarrow currentIndex + 1$
        end if
    end for
    return *items*
end function

Algorithm 5 Fetching items that a user has purchased

function FETCHMYNFTS
    Returns array of MarketItem
    $totalItemCount \leftarrow$ current value of $_tokenIds$
    $itemCount \leftarrow 0$
    $currentIndex \leftarrow 0$
    for $i \leftarrow 0$ **to** $totalItemCount - 1$ **do**
        if owner of $idToMarketItem[i+1]$ is equal to sender's address **then**
            $itemCount \leftarrow itemCount + 1$
        end if
    end for
    Create a new array *items* of type MarketItem with length $itemCount$
    for $i \leftarrow 0$ **to** $totalItemCount - 1$ **do**
        if owner of $idToMarketItem[i+1]$ is equal to sender's address **then**
            $currentId \leftarrow i+1$
            $currentItem \leftarrow idToMarketItem[currentId]$
            $items[currentIndex] \leftarrow currentItem$
            $currentIndex \leftarrow currentIndex + 1$
        end if
    end for
    return *items*
end function

Algorithm 6 Zero-knowledge proof

```
Struct Report: OrgID, validUntil, RedeableAmt, hasCondA, hasCondB, hasCondC
Struct Requirements: OrgID, verifyTime, minRedeableAmt, maxRedeableAmt, allowCondA, allowCondB, allowCondC

function HASHREPORT(report, reqToCheck)
    return hash(report.OrgID, report.validUntil, report.RedeableAmt,
                report.hasCondA, report.hasCondB, report.hasCondC,
                reqToCheck.OrgID, reqToCheck.verifyTime,
                reqToCheck.minRedeableAmt, reqToCheck.maxRedeableAmt,
                reqToCheck.allowCondA, reqToCheck.allowCondB,
                reqToCheck.allowCondC)
end function

procedure VERIFYORGANIZATION(report, reqToCheck)
    Step 1: Verify OrgID
    Assert report.OrgID == reqToCheck.OrgID
    Step 2: Verify validUntil
    Assert currentTime ≤ report.validUntil
    Assert reqToCheck.verifyTime ≤ report.validUntil
    Step 3: Verify RedeableAmt
    Assert report.RedeableAmt ≥ reqToCheck.minRedeableAmt
    Assert report.RedeableAmt ≤ reqToCheck.maxRedeableAmt
    Step 4: Verify Conditions
    if reqToCheck.allowCondA then
        Assert report.hasCondA == TRUE
    end if
    if reqToCheck.allowCondB then
        Assert report.hasCondB == TRUE
    end if
    if reqToCheck.allowCondC then
        Assert report.hasCondC == TRUE
    end if
    Step 5: Generate Verification Hash
    verificationHash = HASHREPORT(report, reqToCheck)
    Step 6: Store Verification Status
    State.verifiedReqHash[verificationHash] = TRUE
    return verificationHash
end procedure
```

The description of the algorithms developed in our framework is given below:

Algorithm 1 is about minting waste and listing it in the marketplace. It creates a token and returns the token ID and the price. Based on this, the createMarketItem function is called upon, as discussed in Algorithm 2.

Algorithm 2 creates a market item with the following information: tokenID, sender's address, contract's address, and price.

Algorithm 3 transfers ownership of the item as well as funds between the parties. This function, named createmarketSale, stores the sender's address and transfers the token from the contract's address to the sender's address. It transfers the payment to the seller.

Algorithm 4 fetches the unsold marketItems function and keeps track of the unsold items in the marketplace. Here, if the owner of the marketItem is equal to the contract's address, then the counter is increased, and the item is fetched based on the ID.

Algorithm 5 fetches the items that the user has purchased. Here, if the owner of the marketItem is verified with the sender's address, then the item is fetched.

Algorithm 6 shows the zero-knowledge proof (ZKP) algorithm.

Algorithm 7 Dynamic or repetative bidding using Walrasian equilibrium.

function AUCTION Input:
N = Number of companies and M = Number of buyers
$Q_i = [Q_1, Q_2, \ldots, Q_N]$ = Supply for each company
$B_j = [B_1, B_2, \ldots, B_M]$ = Budgets for each buyer
$C_i = [C_1, C_2, \ldots, C_N]$ = Reserve prices for each company
$U_j(Q_{ij})$ = Utility function for buyer j
α = Step size for price adjustment and ϵ = Convergence threshold
max_iterations = Maximum number of iterations
Output: P_{ij} = Final price matrix and Q_{ij} = Final allocation matrix
Steps:
Step 1: Initialization
Initialize price matrix P_{ij} with reserve prices C_i:

$$P_{ij}^0 = C_i \quad \text{for all } i, j$$

Initialize iteration counter: $t = 0$
Initialize excess demand $Z_i^t = [0, 0, \ldots, 0]$ (size N)
Step 2: Iterative Adjustment

Repeat until convergence or maximum iterations:

1. Compute demand $D_{ij}(P_{ij}^t)$ for each buyer j and company i:

$$D_{ij} = \arg\max_{Q_{ij}} \left(U_j(Q_{ij}) - P_{ij}^t \cdot Q_{ij} \right)$$

$$\text{Subject to:} \quad Q_{ij} \leq \frac{B_j}{P_{ij}^t} \quad \text{(Budget constraint)}$$

2. Calculate excess demand Z_i^t for each company i:

$$Z_i^t = \sum_{j=1}^{M} D_{ij} - Q_i$$

3. Update prices P_{ij}^t for all i, j:

$$P_{ij}^{(t+1)} = P_{ij}^t + \alpha \cdot Z_i^t$$

4. Check convergence:

$$\text{If } |Z_i^t| \leq \epsilon \quad \text{for all } i: \quad \text{Break the loop}$$

5. Increment iteration counter: $t = t + 1$

Step 3: Final Allocation

For each $i \in [1, N]$:
For each $j \in [1, M]$:

$$Q_{ij} = D_{ij}(P_{ij}^t)$$

Step 4: Output Results Return final price matrix P_{ij} and allocation matrix Q_{ij}
end function

The verification process is carried out using the process of zero-knowledge proofs, which allows companies to be verified without revealing their private information to administrators, who set the criteria. Zero-knowledge proofs are an efficient technique to minimize the amount of information transferred between the prover and the verifier in a cryptographic protocol. The ZKP algorithm has a struct report with the following

fields: OrgID (organization ID), valid until (expiry or validation time), RedeableAmt (redeemable amount), hasCondA, hasCondB, hasCondC (conditions or flags for additional constraints). The constraints are as follows: OrgID (organization ID to check against), verifyTime (time of verification), minRedeableAmt and maxRedeableAmt (limits on the redeemable amount), and allowCondA, allowCondB, allowCondC (allowed conditions for flexibility in validation).

Functions: hashReport(report: Report): Computes a hash value based on the contents of a report. The hash serves as an immutable, verifiable representation of the report.

VERIFYORGANIZATION (report: Report, reqToCheck: Requirements): Validates whether a report satisfies a set of requirements using a zero-knowledge proof approach.

Validates the report using the requirements by:

1. Asserting conditions like matching OrgID (OrgID field in the report matches the OrgID specified in the requirements (reqToCheck)), validUntil time constraints (ensures that the report is not expired and is valid for the requested verification period), and redeemable amount ranges (ensures that only reports with an acceptable amount of redeemable value are validated).
2. Three conditional checks verify whether the specific conditions in the report align with the allowable conditions in the requirements (hasCondA, hasCondB, hasCondC). This ensures that any extra conditions required for validation (e.g., specific certifications, flags) are satisfied.
3. Computing a final verification hash of both the report and the requirements. The hash acts as a unique proof of verification, encapsulating all the validated constraints without exposing the raw data. This is critical for privacy in a zero-knowledge proof system.
4. Storing the hash in State.verifiedReqHash to mark the request as verified.

In Algorithm 7, the verified organizations participate in the auction process. In our framework, a Walrasian equilibrium is used for iterative price adjustments in dynamic bidding within an auction-based waste trading system. The algorithm is given here.

4. Implementation and Result Discussion

4.1. Implementation

The utilization of zkEVM blockchain technology has enabled the process of waste management and the involvement of private organizations in recycling the waste generated, while prioritizing security measures. The proposed architectural framework is developed through the use of zkEVM, Node.Js, Ether.Js, Web3Modal, Wallet Connect SDK, and NFT.storage API endpoints. The development of decentralized applications requires the implementation of Ethereum, a secure, open-source, scalable, and programmable blockchain platform that offers smart contract development functionalities. The designed and implemented Greenlink framework represents the process of user-generated waste management by verified companies through NFTs. It also adds a layer of security by keeping users, administrators (who set criteria), and organizational identities private. This framework has the potential to introduce a new way of waste management, where trust and privacy are backed up by auction-based NFT trading.

The integration of the various tools and technologies used and their integration within our proposed framework is listed here:

1. Ether.js: Interacts with the Ethereum blockchain. It also provides functionalities to manage wallets, handle transactions, etc.
2. zkEVM: The proposed architectural framework is developed using zkEVM. It facilitates cost-effectiveness and scalability in blockchain transactions and enhances

waste management operations. Quicker transaction processing is facilitated by the duo zero-knowledge rollups (zk-rollups) and the Ethereum Virtual Machine (EVM), ensuring lower fees and that security is also not compromised. The zk-rollups further help in waste management transaction confirmation. The zkEVM and EVM efficiently handle dynamic bidding and NFT issuance for waste-tracking operations.

3. Wallet Connect allows decentralized applications to connect and interact with cryptocurrency wallets securely.
4. Web3Modal is a library that makes connecting decentralized applications to different cryptocurrency wallets simple.
5. NFT.storage: The decentralized storage functionalities for NFT metadata are provided by the NFT's highly dependable storage service. It also helps in tamper-proof record storage for waste management (like origin and recycling history). This enables the stakeholders to confirm the waste sources (i.e., traceability) and the process of recycling. NFT.storage is a specialized service designed to provide decentralized storage functionalities for non-fungible tokens. This also makes for very cost-effective data management.
6. Coinbase Wallet SDK enables users to connect the Coinbase Wallet to Greenlink.
7. Hyperledger Caliper is a performance evaluation benchmarking tool.

For deployment on the blockchain, the address of the current model's contract owner is used. The private key is given in checksum form. To access and use it, the user can mint the token. The Ethereum nodes hold the private key. In the caliper, the signed transactions and the data are sent to the node when the private key is available for the address of the model's contract owner.The address of the benchmark invokes the methods in the contract .The network_config file accesses the from_address and its private_key with a password to invoke the methods in the caliper benchmark.

The caliper adapter hosts the successful transactions over the network along with the block count. The JSON object is given by the config_list to be deployed on the network prior to executing the caliper benchmark. JSON entries for the createMarketItem, createMarketSale, createToken, fetchItemListed smart contracts of our model are specified, and this enables the invocation of the methods on the contracts.

4.2. Result Discussion

The proposed NFT marketplace model's performance is evaluated using the Hyperledger Caliper, and the Polygon blockchain is used over Ethereum network. The OS is Ubuntu 18.04. The various tools are listed in Table 3.

Performance metrics like the speed of transactions, tokenization (which ensures precision and reliability), and consumption of resources (CPU and memory usage) are monitored. These are evaluated based on the transaction rates for the network's scalability and efficiency measurement. This helps us to evaluate and assess the real-world performance of our proposed model.

The benchmarking of each step is performed by providing an input size (load) of around 5000 transactions. Table 4 shows the overall performance of the proposed model by Hyperledger Caliper, which reflects the fact that every transaction was successfully performed over the Ethereum network. This table shows the send rate, latency, and throughput of the proposed model's four essential smart contracts.

Table 3. List of tools/software used with their version.

Sl#	Requirements	Specifications
1	React	v17.0.0
2	Vite	v2.9.9
3	Ethers.js	v5.6.6
4	Web3Modal	v1.9.7
5	Coinbase Wallet SDK	v3.1.0
6	Solidity	v0.8.7
7	NFT.storage	v6.2.0
8	Wallet Connect	v1.7.8
9	NPM	v10.5.0
10	Vs Code	v1.92.0

The current model's performance is tested with a total transaction count of 5000. The input transaction rates are 1000, 2000, 3000, 4000, and 5000 per second. The CreateMarketItem, createMarketSale, createToken, and fetchItemListed functions' performance is noted. The number of workers varied from 1–5. A total of 15 evaluation runs were conducted (i.e., for workers 1–5 for every transaction rate, i.e., 1000, 2000, 3000, 4000, and 5000 per second). The performance data are summarized in Tables 4 and 5. The various performance metrics like latency, throughput, memory usage, CPU usage, traffic in/out, and disk read/write are plotted in the graphs in the results subsection. The utilization of resources like CPU, memory, throughput, and traffic flowing in and out are shown in Table 5.

Table 4. Send rate, average latency, and throughput of the four functions used in our model.

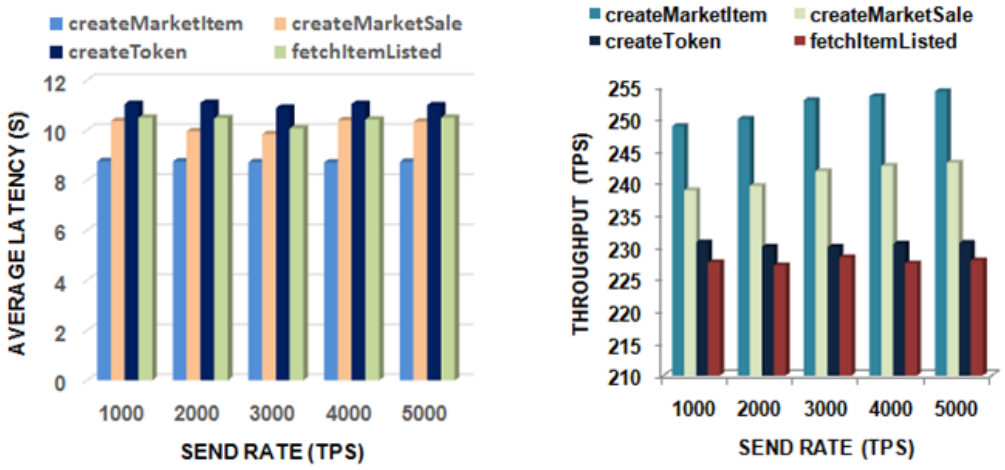
Function Name	Txns	Success	Send Rate (TPS)	Avg Latency (s)	Throughput (TPS)
create Market Item	5000	5000	1000	8.76	248.8
	5000	5000	2000	8.75	249.9
	5000	5000	3000	8.72	252.8
	5000	5000	4000	8.71	253.4
	5000	5000	5000	8.74	254.2
create Market Sale	5000	5000	1000	10.37	238.8
	5000	5000	2000	9.96	239.5
	5000	5000	3000	9.84	241.8
	5000	5000	4000	10.4	242.6
	5000	5000	5000	10.34	243.1
create Token	5000	5000	1000	11.05	230.7
	5000	5000	2000	11.09	230
	5000	5000	3000	10.89	230
	5000	5000	4000	11.05	230.5
	5000	5000	5000	11.02	230.6
fetch Item Listed	5000	5000	1000	10.51	227.6
	5000	5000	2000	10.48	227.1
	5000	5000	3000	10.07	228.4
	5000	5000	4000	10.43	227.4
	5000	5000	5000	10.51	227.9

Table 5. CPUT utilization, memory utilization, and traffic in/out of the four functions with respect to the send rate 1000 to 5000 TPS.

Function Name	Send Rate (TPS)	CPU% (max)	CPU% (avg)	Memory (max) [MB]	Memory (avg) [MB]	Traffic In [MB]	Traffic Out [MB]
create Market Item	1000	93.9512	51.18625	791	745	12.9	28.2
	2000	92.9511	50.5484	825	772	12.8	26.9
	3000	91.2542	49.2547	888	780	13.1	28.9
	4000	90.9087	47.2548	902	826	12.9	27.8
	5000	93.2588	47.5457	988	872	12.9	28.1
create Market Sale	1000	25.7462	4.3125	989	784	9.95	31.2
	2000	26.5871	4.4154	992	810	9.81	30.2
	3000	27.5844	4.7871	993	809	9.73	30.2
	4000	28.5857	4.7985	987	842	9.93	31.2
	5000	29.9899	4.8215	991	859	10.02	31.1
create Token	1000	28.0651	4.6914	22.065	4.69	12.1	32.5
	2000	28.4362	4.7678	23.854	4.76	12.1	32.6
	3000	28.8981	4.8125	23.754	5.01	11.9	31.6
	4000	29.8785	4.9254	23.896	4.81	12.1	32.5
	5000	29.8025	4.9891	24.021	4.91	12.2	32.4
fetch Item Listed	1000	23.2541	4.1254	22.854	4.6535	10.3	31.5
	2000	24.8765	4.3658	23.459	4.1625	10.2	31.4
	3000	26.5898	4.4989	23.898	4.7965	10.3	32.2
	4000	27.0054	4.5175	23.878	4.5175	10.2	31.3
	5000	28.4575	4.7585	23.587	4.7585	10.3	31.4

Table 5 represents the analysis of the latency for operations like createMarketItem, createMArketSale, createToken, and fetchItemListed, respectively. These functions were measured at 1000, 2000, 3000, 4000, and 5000 transactions per second (TPS), respectively. The latency details in Table 4 are plotted in Figure 2a, which reveals that average latency is relatively high for the CreateToken function (deep blue color), whereas it is lower for the createMarketItem function (light blue color). Another trend also reveals that as the send rate (transactions per second) increases, the average latency gradually decreases. The average latency reaches 8.71 s when the workers are set at 3. The average latency reaches a maximum of 8.76 s when the no. of workers = 4 for the CreateMarketItem function. Similarly, the average latency ranges from a maximum of 10.37 s to a minimum of 9.84 s. The createToken function's average latency range ranges from a minimum of 10.89 s to a maximum of 11.05 s. For the function fetchItemListed, the maximum average latency is 10.51 s and the minimum is 10.07 s.

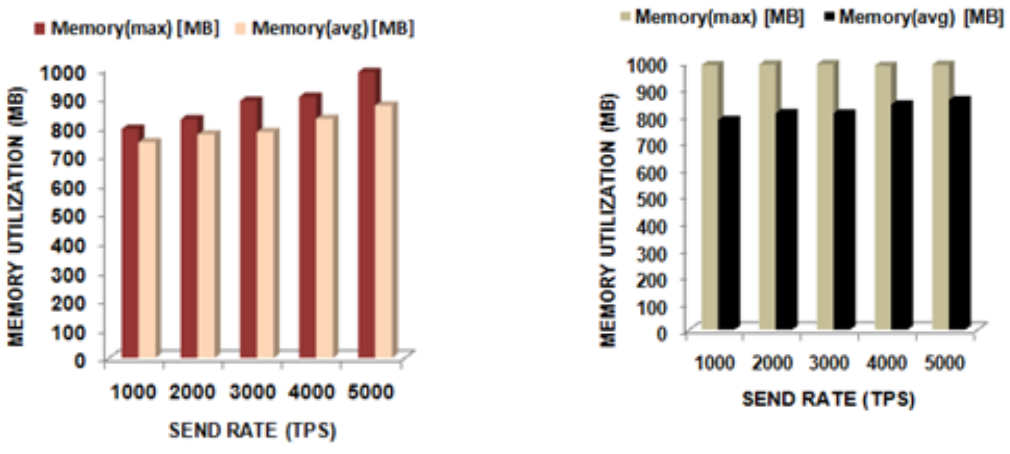
The graph in Figure 2b shows the throughput for createMarketItem, createMArketSale, createToken, and fetchItemListed, respectively, when the transaction per second is varied from 1000 to 5000, as shown in Table 4. In Figure 2b, the createToken function has a lower throughput compared to the other three tokens (deep red color in Figure), and the createMarketItem has the highest throughput (light blue color) of all the functions for all the rates of transactions (1000 to 5000). The graphs in Figure 2a,b have been drawn with different color codes to differentiate the four functions, and they are grouped together for each of the transaction rates 1000–5000 TPS, respectively.



(**a**) (**b**)

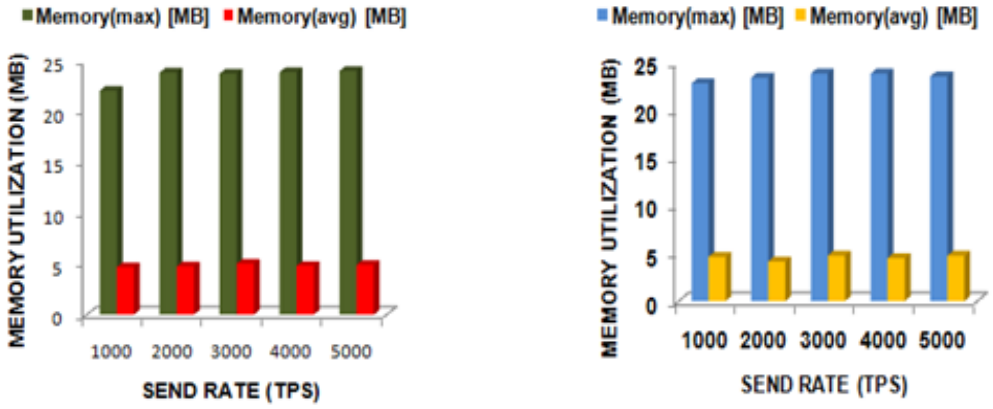
Figure 2. (**a**) Average latency; (**b**) throughput. (**a**) shows the average latency of the solidity functions createMarketItem, createMarketSale, createToken, and fetchItemListed; w.r.t send rate (in TPS). (**b**) shows the throughput of the solidity functions createMarketItem, createMarketSale, createToken, and fetchItemListed; w.r.t send rate (in TPS).

Figure 3a shows the graph of the memory utilization in MB versus the send rate (1000–5000 TPS) of the createMarketItem function. The graph shows that the maximum memory utilization is 791 MB when the send rate is 1000 TPS, and it is 988 MB when the send rate is 5000 TPS. For the createMarketSale function, Figure 3b shows the maximum memory utilization is 989 MB when the send rate is 1000 TPS, and it is 991 MB when the send rate is 5000 TPS. For the createToken function, as shown in Figure 3c, the maximum memory utilization is 22.0 MB when the send rate is 1000 TPS, and it is at 24.0 MB when the send rate is 5000 TPS. In the fetchItemListed function, the graph in Figure 3d shows the maximum memory utilization is 22.8 MB when the send rate is 1000 TPS, and it is 23.5 MB when the send rate is 5000 TPS. The maximum memory consumption for the createToken and fetchItemListed functions is low for all the transaction rates in the range of 1000–5000 TPS (22.0 MB to 24.0 MB for both functions).



(**a**) (**b**)

Figure 3. *Cont.*



(**c**) (**d**)

Figure 3. (**a**) Memory utilization of createMarketItem function; (**b**) memory utilization of createMarketSale function; (**c**) memory utilization of createToken function; (**d**) memory utilization of fetchItemListed function. (**a**) shows the memory utilization (maximum memory used and Average memory utilized) of the smart contract—createMarketItem versus send rate (in TPS). (**b**) shows the memory utilization (maximum memory used and Average memory utilized) of the smart contract—createMarketSale versus send rate (in TPS). (**c**) shows the memory utilization (maximum memory used and Average memory utilized) of the smart contract—createToken versus send rate (in TPS). (**d**) shows the memory utilization (maximum memory used and Average memory utilized) of the smart contract—fetchItemListed versus send rate (in TPS).

Figure 4a shows the graph of the CPU utilization in % versus the send rate (1000–5000 TPS). For the createMarketItem function, the graph shows that the maximum CPU utilization ranges from 90.90% (4000 TPS) up to 93.95% (1000 TPS). For the createMarketSale function, Figure 4b shows the maximum memory utilization is in the range of 26.58% (2000 TPS) up to 29.98% (5000 TPS). For the function createToken, Figure 4c shows the maximum memory utilization ranges from 28.0% (1000 TPS) up to 29.8% (5000 TPS). For the fetchItemListed function, Figure 4d shows the maximum memory utilization ranges from 23.2% (1000 TPS) up to 28.4% (5000 TPS). In summary, the memory utilization function is highest for createMarketItem, within the range of 90.9–93.9%. However, for the other three functions—createMarketSale, createToken, and fetchItemListed—the CPU utilization is low, within the range of 25.7–29.9%.

The memory utilization graph shown in Figure 3 shows that for the createMarketItem and createMarketSale functions, memory utilization remains high as the send rate increases. Both maximum and average memory utilization are within the range of 800–1000 MB, indicating that this function is resource-intensive and incurs significant processing tasks. The createToken and fetchItemListed functions have much lower memory usage, with a maximum memory of 25 MB and an average memory of 5 MB, which indicates efficiency in creating tokens with low resource demand, as token creation and tracking waste units is a lighter task. The CPU utilization graph, as shown in Figure 4, depicts that for the createMarketItem function, the maximum CPU utilization percentage is high, nearing 90–100%, regardless of the send rate. Average CPU consumption is significantly lower at around 40–50%, which indicates that creating market items for NFTs is CPU-intensive, as it handles complex processes like minting NFTs for waste management and recording metadata. However, the load cannotbe sustained continuously. The CPU utilization of the createMarketSale and createToken functions shows that the minimum CPU consumption is around 25–30%, with an average CPU utilization % below 10%, which suggests that facilitating market sales of NFTs is less CPU-demanding compared to the createMarketItem function. In addition, the creation of tokens (representing unique identifiers for waste items) appears to be lightweight, requiring relatively few CPU

resources, while the fetchItemListed function also requires minimal CPU resources; this makes the system efficient in token generation and retrieving data, which is critical for large-scale waste management scenarios. The proposed model design is scalable, and this is shown by the fact that, for all the function designs, CPU utilization and memory consumption remain consistent across the send rates, which is vital for blockchain-based waste management systems as the volume of transactions increases. The memory utilization and CPU utilization graphs for the the createMarketItem function show high resource utilization, as the MarketItem is likely a struct with multiple fields and, as we assign a new MarketItem to the mapping, it consumes memory (storage) for each of these fields. The id-market item is a key value pair in the mapping and requires additional storage. The storage consumption increases significantly as the mapping grows with more entries. Payable addresses, like payable(msg.sender) and payable(address(this)), are stored in the struct, called MarketItem, which adds additional data to the structure, increasing memory usage. For many token listings, if this function is called multiple times, the storage usage increases.

As a result of the above-mentioned processing, it impacts scalability by gas cost, throughput reduction, and network congestion. Hence, off-chain data storage, sharding, and lazy minting will be preferred in our future work.

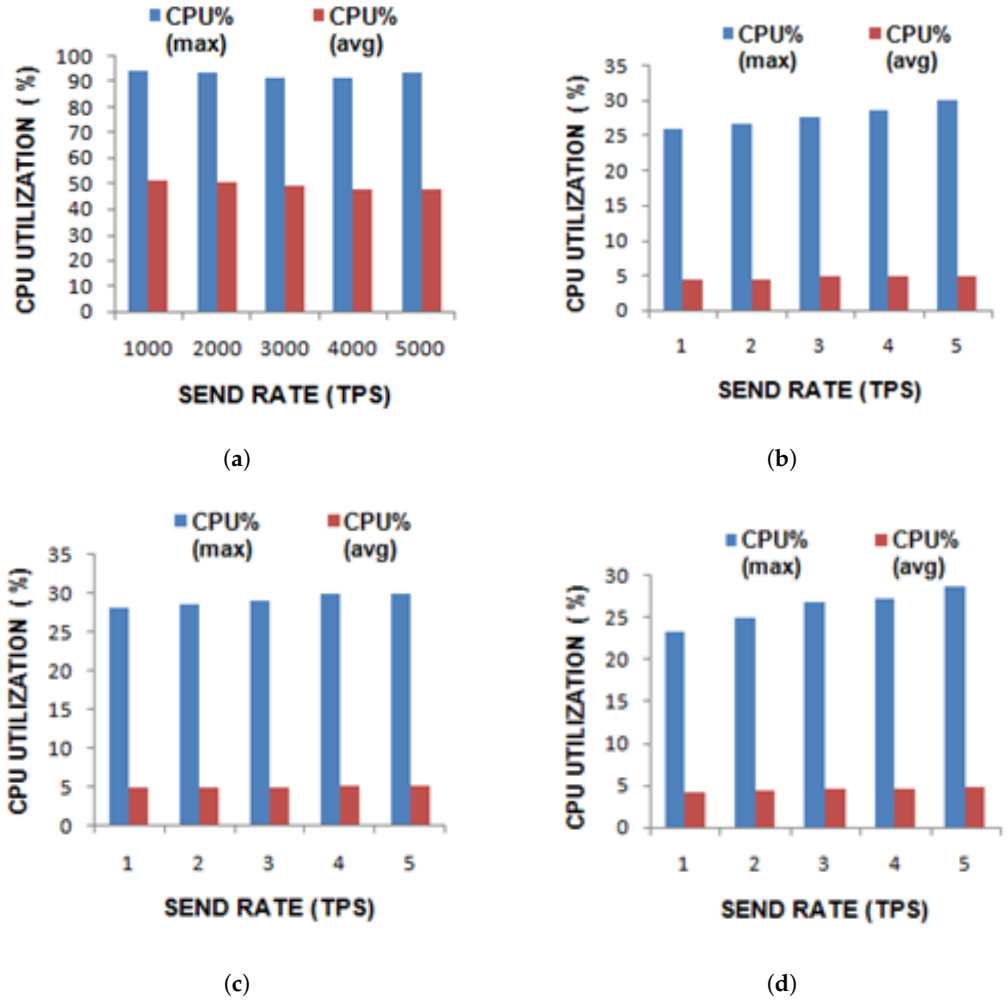


Figure 4. (**a**) CPU utilization of createMarketItem function; (**b**) CPU utilization of createMarketSale function; (**c**) CPU utilization of createToken function; (**d**) CPU utilization of fetchItemListed function. (**a**) shows the CPU utilization of the smart contract—createMarketItem versus send rate (in TPS). (**b**) shows the CPU utilization of the smart contract—createMarketSale versus send rate (in TPS). (**c**) shows the CPU utilization (maximum memory used and average memory utilized) of the smart contract—createToken versus send rate (in TPS). (**d**) shows the CPU utilization of the smart contract—fetchItemListed versus send rate (in TPS).

5. Conclusions

In this article, we have proposed a blockchain-enabled non-fungible token and dynamic bidding using a Walrasian equilibrium to achieve iterative price adjustments for sustainable waste management. We used the ZKP validity mechanism, an innovative framework based on NFTs that facilitates the minting of waste and listing it in the marketplace, as well as the creation of the market item, transferring ownership of funds and items between parties, fetching NFTs, and returning items listed by users. With a seamless interaction, an elegant GUI was developed, with the features of reliability, transparency, and security. Additionally, we conducted a comparative analysis, evaluating performance using the Hyperledger Caliper tool, examining transaction rates, and analyzing the resulting metrics. Analysis of the latency and throughput was performed for the token operations—CreateMarketItem, createMarketSale, createToken, and fetchItemListed. A detailed performance evaluation of our built model was also conducted based on the maximum and minimum CPU utilization, memory utilization, and traffic in/out of each of the token operations (CreateMarketItem, createMarketSale, createToken, fetchItemListed) for each of the send rates from 1000 TPS to 5000 TPS. The results are plotted, proving the efficacy of the current proposed model. The results show that the maximum average latencies of CreateMarketItem, createMarketSale, createToken, and fetchItemListed are 8.76 s, 10.37 s, 11.09 s, and 10.51 s, respectively, and the minimum average latencies are 8.71 s, 9.84 s, 10.89 s, and 10.51 s, respectively. Average latency is relatively high for the CreateToken function, whereas it is lower for the createMarketItem function. Another trend also reveals that as the send rate (transactions per second) increases, the average latency comes down gradually. createMarketItem has the highest throughput, averaging at 251.5 s, and the createToken function has the lowest throughput, averaging at 32.5 s. The average memory consumption of createMarketItem and createMarketSale lies within the range of 772–872 MB for the range of TPS from 1000 to 5000. The average memory consumption of createToken and fetchItemListed lies within the range of 4.6 MB to 5.01 MB for the range of TPS from 1000 to 5000. The traffic in lies within the range of 9.93 to 13.01 and the traffic out ranges from 28.1 MB to 32.6 MB for all the four tokens analyzed. The % maximum CPU utilization of the four functions lies in the range of 23.2 MB to 29.9 MB for the three functions createMarketSale, createToken, and fetchItemListed, and for the createMarketItem, it is 92.95 MB. This is a new dimension in the performance aspect of non-fungible token-enabled sustainable waste management approaches using the zero-knowledge proof (ZKP) validity mechanism. Sustainable waste management is very important, and the advantages of the NFT technology of blockchain, which has also been integrated with the based verification of parties method, have paved the way for a unique dimension in this arena. This promises to bring new developments in waste management and asset ownership through NFTs with the advantages of blockchain.

The limitations of the proposed research would be in providing the participants with real-time data on the environmental impact of their contributions, such as the amount of CO_2 saved or resources recycled. This gives a sense of purpose and also motivates further engagement. Another limitation would be the dynamic adjustment of resource allocation using a load balancer based on real-time network activity, which ensures smooth operation during peak loads. Future work for this research should aim to expand the Greenlink framework's applicability to adjacent domains, like supply chain management, recycling, and carbon trading, to demonstrate its versatility and further validate its utility. Another ongoing work will aim to developa dynamic load balancer for dynamically adjusting the resource allocation.

Assumption 2. *The allocation of the waste quantity cannot exceed available supply or budgetary limits. Each organization maximizes its utility, while each company maximizes its revenue. Prices are determined based on bidding values and reserve prices.*

Author Contributions: Conceptualization, C.S.K., A.B.P. and A.P.S.; Methodology, C.S.K. and K.H.K.R.; Software, C.S.K. and A.B.P.; Validation, C.S.K.; Formal analysis, A.P.S.; Resources, A.B.P.; Writing—original draft, C.S.K.; Writing—review & editing, K.H.K.R.; Supervision, A.P.S.; Funding acquisition, K.H.K.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Khiem, H.G.; Khanh, H.V.; Huong, H.L.; Quy, T.L.; Phuc, T.N.; Ngan, N.T.K.; Triet, M.N.; Bang, L.K.; Trong, D.P.N.; Hieu, M.D.; et al. Implementing a blockchain, smart contract, and NFT framework for waste management systems in emerging economies: An investigation in Vietnam. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 985–996.
2. Bułkowska, K.; Zielińska, M.; Bułkowski, M. Implementation of blockchain technology in waste management. *Energies* **2023**, *16*, 7742. [CrossRef]
3. Jiang, P.; Zhang, L.; You, S.; Van Fan, Y.; Tan, R.R.; Klemeš, J.J.; You, F. Blockchain technology applications in waste management: Overview, challenges and opportunities. *J. Clean. Prod.* **2023**, *421*, 138466. [CrossRef]
4. Tian, Z. Post-merge carbon footprint analysis and sustainability in the NFT Art Market. *Arts* **2023**, *12*, 211. [CrossRef]
5. Mielcarek, A.; Kłobukowska, K.; Rodziewicz, J.; Janczukowicz, W.; Bryszewski K, Ł. Water Nutrient Management in Soilless Plant Cultivation versus Sustainability. *Sustainability* **2023**, *16*, 152. [CrossRef]
6. Chiacchio, F.; D'Urso, D.; Oliveri, L.M.; Spitaleri, A.; Spampinato, C.; Giordano, D. A non-fungible token solution for the track and trace of pharmaceutical supply chain. *Appl. Sci.* **2022**, *12*, 4019. [CrossRef]
7. Chiquito-Contreras, R.G.; Hernandez-Adame, L.; Alvarado-Castillo, G.; Martínez-Hernández, M.d.J.; Sánchez-Viveros, G.; Chiquito-Contreras, C.J.; Hernandez-Montiel, L.G. Aquaculture—Production System and Waste Management for Agriculture Fertilization—A Review. *Sustainability* **2022**, *14*, 7257. [CrossRef]
8. El-Rayes, N.; Chang, A.; Shi, J. Plastic Management and Sustainability: A Data-Driven Study. *Sustainability* **2023**, *15*, 7181. [CrossRef]
9. Triet, M.N.; Khanh, H.V.; Huong, H.L.; Khiem, H.G.; Phuc, T.N.; Ngan, N.T.K.; Quy, T.L.; Bang, L.K.; Trong, D.P.N.; Hieu, M.D.; et al. Leveraging Blockchain, Smart Contracts, and NFTs for Streamlining Medical Waste Management: An Examination of the Vietnamese Healthcare Sector. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 958–970.
10. Ahmad, R.W.; Salah, K.; Jayaraman, R.; Yaqoob, I.; Omar, M. Blockchain for waste management in smart cities: A survey. *IEEE Access* **2021**, *9*, 131520–131541. [CrossRef]
11. Ma, X.; Yuan, H.; Du, W. Blockchain-Enabled Construction and Demolition Waste Management: Advancing Information Management for Enhanced Sustainability and Efficiency. *Sustainability* **2024**, *16*, 721. [CrossRef]
12. Steenmans, K.; Taylor, P.; Steenmans, I. Blockchain technology for governance of plastic waste management: Where are we? *Soc. Sci.* **2021**, *10*, 434. [CrossRef]
13. Al-Farsi, S.; Rathore, M.M.; Bakiras, S. Security of blockchain-based supply chain management systems: Challenges and opportunities. *Appl. Sci.***2021**, *11*, 5585. [CrossRef]
14. Soori, M.; Jough, F.K.G.; Dastres, R.; Arezoo, B. Blockchains for Industrial Internet of Things in Sustainable Supply Chain Management of Industry 4.0, A Review. *Sustain. Manuf. Serv. Econ.* **2024**, *3*, 100026. [CrossRef]
15. Aslam, J.; Saleem, A.; Khan, N.T.; Kim, Y.B. Factors influencing blockchain adoption in supply chain management practices: A study based on the oil industry. *J. Innov. Knowl.* **2021**, *6*, 124–134. [CrossRef]
16. Dietrich, F.; Ge, Y.; Turgut, A.; Louw, L.; Palm, D. Review and analysis of blockchain projects in supply chain management. *Procedia Comput. Sci.* **2021**, *180*, 724–733. [CrossRef]
17. Duan; Keru; Pang, G.; Lin, Y. Exploring the current status and future opportunities of blockchain technology adoption and application in supply chain management. *J. Digit. Econ.* **2023**, *2*, 244–288. [CrossRef]
18. Dudczyk, P.; Dunston, J.; Crosby, G.V. Blockchain Technology for Global Supply Chain Management: A Survey of Applications, Challenges, Opportunities and Implications. *IEEE Access* **2024**, *12*, 70065–70088. [CrossRef]

19. Marin, O.; Cioara, T.; Toderean, L.; Mitrea, D.; Anghel, I. Review of Blockchain Tokens Creation and Valuation. *Future Internet* **2023**, *15*, 382. [CrossRef]
20. Rodinko, M.; Oliynykov, R.; Nastenko, A. Decentralized Proof-of-Burn auction for secure cryptocurrency upgrade. *Blockchain Res. Appl.* **2024**, *5*, 100170. [CrossRef]
21. Cui, S.; Xu, S.; Hu, F.; Zhao, Y.; Wen, J.; Wang, J. A consortium blockchain-enabled double auction mechanism for peer-to-peer energy trading among prosumers. *Prot. Control Mod. Power Syst.* **2024**, *9*, 82–97. [CrossRef]
22. Kadadha, M.; Mizouni, R.; Singh, S.; Otrok, H.; Ouali, A. ABCrowd an auction mechanism on blockchain for spatial crowdsourcing. *IEEE Access* **2020**, *8*, 12745–12757. [CrossRef]
23. Agarwal, S., Kapoor, S., Jain, A. Double Auction Mechanism Based Energy Trading in Blockchain Enabled Smart Microgrid. In Proceedings of the 2024 IEEE 3rd International Conference on Electrical Power and Energy Systems (ICEPES), Faridabad, India, 21–22 June 2024; pp. 1–7.
24. Liu, T.; Wu, J.; Chen, L.; Wu, Y.; Li, Y. Smart contract-based long-term auction for mobile blockchain computation offloading. *IEEE Access* **2020**, *8*, 36029–36042. [CrossRef]
25. Yang, X.; Li, W. A zero-knowledge-proof-based digital identity management scheme in blockchain. *Comput. Secur.* **2020**, *99*, 102050. [CrossRef]
26. Wali, H.; Kulkarni, V.; Ganiger, R.; Iyer, N.C. ZKAV: Zero Knowledge proof for AV. *Procedia Comput. Sci.* **2024**, *237*, 891–898. [CrossRef]
27. Soewito, B.; Marcellinus, Y. IoT security system with modified Zero Knowledge Proof algorithm for authentication. *Egypt. Inform. J.* **2021**, *22*, 269–276. [CrossRef]
28. Zhou, L.; Diro, A.; Saini, A.; Kaisar, S.; Hiep, P.C. Leveraging zero knowledge proofs for blockchain-based identity sharing: A survey of advancements, challenges and opportunities. *J. Inf. Secur. Appl.* **2024**, *80*, 103678. [CrossRef]
29. Kuznetsov, O.; Rusnak, A.; Yezhov, A.; Kanonik, D.; Kuznetsova, K.; Karashchuk, S. Enhanced Security and Efficiency in Blockchain with Aggregated Zero-Knowledge Proof Mechanisms. *IEEE Access* **2024**, *12*, 49228–49248. [CrossRef]
30. Xu, X. Zero-knowledge proofs in education: A pathway to disability inclusion and equitable learning opportunities. *Smart Learn. Environ.* **2024**, *11*, 7. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.