Available online to ejournal.pnc.ac.id

Journal of Innovation Information Technology and Application

Journal Page is available to https://ejournal.pnc.ac.id/index.php/jinita

Blockchain Based Donation Management in Disaster Response

Ankit K C

Department of Computer Science, C.V. Raman Global University, Bhubaneswar, India

email: ankitkc908@gmail.com

ARTICLE INFO

Article history:
Received 13 March 2024
Revised 15 May 2024
Accepted 29 May 2024
Available online 28 June 2024

Keywords:
Transaction
Corruption
Traceability
Accountability
Immutability

IEEE style in citing this article:
A. K C, "Blockchain Based Donation Management in Disaster Response," Journal of Innovation Information Technology and Application (JINITA), vol. 6, no. 1, pp. 45–59, Jun. 2024.

ABSTRACT

Both natural and man-made disaster leave thousands of people in vulnerable and in need for essential aid. While individuals generously donate resources, traditional donation management systems suffer from limitations. Centralized control, opaque transactions, and potential corruption often hinder aid delivery and leave victims in despair. This paper proposes a novel Ethereum blockchain-based system for transparent and secure disaster donation management. Utilizing smart contracts, the system ensures traceability, accountability, and immutability of donations, empowering donors and fostering trust. This paper presents the system's architecture, detailing its components and interactions through sequence diagrams and algorithms. Additionally, successful testing on the Sepolia Ethereum testnet validates its functionality. To assess its effectiveness, a cost and security analysis is conducted. This blockchain-based framework offers a promising solution for transparent and efficient disaster response, potentially revolutionizing donation management. Further research, particularly on donation allocation optimization within the system holds immense potential for future development.

1. INTRODUCTION

In the aftermath of disasters, both natural and man-made communities across the globe face profound disruption and displacement. Effected people get deprived from access to vital resources like healthcare, food, and shelter [1]. In this critical moment, timely and coordinated disaster response becomes crucial. However, traditional donation management systems often struggle to deliver aid efficiently and transparently, leaving crucial resources stranded and beneficiaries questioning the efficacy of their support.

Imagine a scenario where donated medical supplies, intended to relief for victims of an earthquake, remain lost in an opaque system, their journey untraceable and their impact unknown. This is the harsh reality under the current paradigm, where slow processes, opaque transactions, and vulnerabilities to fraud cast a shadow over the noble act of giving [2]. This necessitates a paradigm shift, a search for innovative solutions that can guarantee transparency, efficiency, and accountability in disaster aid delivery, ensuring every contribution makes a tangible difference.

A study by A. Rawat et al. [3] reveals that large quantities of donated tents, clothing, and food were unused in government warehouses months after the 2015 Nepal earthquake. The study attributed to poor distribution systems, lack of needs assessments, and political interference in donation management. Digital Accountability, 2023 report emphasizes on the crucial role of digital solutions in enhancing transparency and accountability in humanitarian aid through blockchain technology being used for tracking donations and ensuring their efficient delivery to beneficiaries [4]. In a post disaster situation various aid organizations and government agencies start supplying food, water, clothing, medicines, and other emergency relief materials efficiently and quickly to maximize survival rate and continue normalcy [5].

*) Corresponding Author: ankitkc908@gmail.com

However, managing disaster response is not that straight forward. In most disasters, information is scarce between the donor and victims and their coordination rarely exists.

The centralized nature of traditional systems where a single entity controls the entire flow of donations and decision-making structure creates a single point of failure: if the central authority becomes corrupt or incapacitated, the entire system collapses, leaving victims in vulnerable situation [6]. Also, limited transparency restricts both donors and beneficiaries from understanding how many requests are made and resources received, further diminishing trust and accountability.

Blockchain's decentralized nature offers significant advantages in disaster response settings. Unlike centralized systems which are prone to single points of failure, blockchain remains operational even if individual nodes are compromised [6]. Additionally, blockchain facilitates seamless collaboration across disparate organizations, acting as a bridge for immutable data and information sharing [7]. This ensures uninterrupted and resilient coordination among stakeholders. While traditional donation management systems often struggle with opaque processes and vulnerability to fraud, blockchain technology presents a promising avenue for improvement. By leveraging its decentralized ledger, blockchain creates a permanent and immutable record of all transactions, providing transparency for donors and stakeholders.

A study by Simon et al. [8] outlines the critical role of accountability mechanisms in disaster response. It emphasizes the need for clear reporting systems, feedback mechanisms, and participation of affected communities to ensure aid reaches those who need it most. Within a context of blockchain based model for disaster response,[9], [10] proposes a model for disaster aid supply based on blockchain technology. However, these models lack traceability and donor collaboration whereas [11], [12] presents the donation management model which can trace the donation for charity purposes. Current related work demonstrates the advantages and possibilities of blockchain technology applied in the donation and supply chain in disaster response. This study aims to develop and evaluate a novel blockchain-based solution for enhanced donation management in disaster response. The objective is to address the following critical challenges in existing systems.

- Lack of coordination among stakeholders involved in disaster relief efforts.
- Difficulty in ensuring transparent and traceable donation flows from donors to beneficiaries.
- Absence of a decentralized robust system for recording and verifying donation data.

By leveraging the core strengths of blockchain technology, the proposed solution offers a secure, transparent, and auditable framework for donation management in disaster response. Blockchain's tamper-proof nature ensures the security and immutability of donation data, while its distributed ledger allows for real-time traceability throughout the donation cycle. This enhanced accessibility fosters collaboration and information exchange among stakeholders, ultimately contributing to a more efficient and trustworthy disaster response system. By achieving these objectives, this study outlined the following major contributions to blockchain based donation management in disaster response.

- Proposed solution provides security, traceability, immutability, and accessibility of information for effective coordination among stakeholders.
- A smart contract is designed which is capable of handling various activities like donation, request, tracing among disaster-response stakeholders.
- Algorithm of proposed solution along with their full implementation, testing and validation is presented.
- Security and cost analysis to evaluate the performance of the proposed blockchain-based solution is conducted.
- Comparison of proposed solutions with existing blockchain and non blockchain based solutions is done to show the novelty of presented solution.

The remainder of this paper is structured as follows: Section 2 reviews related work in the field, providing context for the proposed solution. Section 3 outlines the architect of proposed blockchain-based donation management system for disaster response. Section 4 presents implementation details with algorithm while Section 5 shows testing and validation of proposed solution. Section 6 offers a comprehensive discussion and evaluation of the proposed solution, followed by conclusion and future work in Section 7.

2. RELATED WORK

Blockchain technology shows potential impact on the effectiveness and transparency of donation management in disaster response. In this section, the related work done in the field of donation and disaster management using blockchain technology is covered.

A study by H. Saleh et al. [13] proposed a private Ethereum blockchain-based solution to automate blood donation management in a manner that is decentralized, transparent, traceable, auditable, private, secure, and trustworthy. A. Almaghrabi et al. [12] proposes a donation traceability framework intended to enable all involved parties to trace the progress of charity donations from the moment they are given by donors to the moment they reach the intended recipients based on public-permissioned blockchain on the Ethereum platform. A paper by H. Wu et al. [14] proposes a charity donation service based on blockchain combining needs of help-seeking, receiving and management user with information release and sharing to alleviate the trust crisis of charity service. However, those solutions are not generalized for disaster response but provide a brief potential of blockchain in donation management.

A. Javadpour et al. [15] proposed disaster response model which reduce response times and ensure the secure and timely distribution of good through the integration of blockchain technology, the Internet of Things (IoT), and the Internet of Everything (Ioe) into the disaster management structure. F. Sabz Ali Pour et al. [9] proposes blockchain technology, the Internet of Things (IOT) and Dynamic Voltage Frequency Scaling (DVFS) algorithm integrated in network-based simulation for advancement in disaster-aids network strategies using smart contract for collaborations.

A. Musamih et al. [16] present an Ethereum blockchain-based approach using smart contracts and decentralized off-chain storage mechanism for efficient product traceability and eliminating the need for intermediaries and provides a secure, immutable history if transaction to all stakeholders. H. Saleh et al. [11] build a donation tracking platform based on blockchain technology which offers transparent accounting of operations and allow donors to monitor where, when and to whom went resources to charity funds.

M. Faherza et al. [10] build an information system in post-disaster management based on blockchain which provide transparent information regarding the data on the good needed and data on goods to be distributed to ensure amount used and not used so that victim affected by disaster can be handled quickly and appropriately. D. Kolhatkar et al. [17] proposed a blockchain system features, such as tamper-proof data storage, automated processes, and decentralized coordination provide a solution to the challenges faced in disaster management and relief efforts.

J. Bhatelia et al. [18] developed a solution based on Hyperledger fabrics that provides anonymity and accountability to the user. Users can donate both items and money through the proposed system and barcode will be generated to facilitate tracking. F. Sunmola et al. [19] paper examines the potential of blockchain technology in ensuring transparency and traceability in disaster relief supply chains. It argues that blockchain's immutable and distributed nature can prevent tampering and provide real-time visibility into the movement of goods.

While existing literature highlights the immense potential of blockchain technology in both donation management and disaster response, it's true that there remains a gap when it comes to a comprehensive and integrated system specifically designed for donation management during disasters. The proposed blockchain-based solution for donation management in disaster response aims to fill that gap.

3. PROPOSED ARCHITECTURE

The proposed architecture is designed to involve key stakeholders seamlessly while ensuring transparency and efficiency as shown in Figure 1. All stake holders which include Donor, Governing Authority, Crisis location and public engage with the system through a front-end software interface. Donors, represented by individual contributors or organizations, and governing authority oversee the entire process ensuring request from true effected locations. Crisis location requests originate from the affected public, indicating specific disaster areas in need. Metamask serves as a connection provider, enabling secure transactions and interactions with the blockchain. On-chain resources encompass smart contracts and the blockchain itself, providing the backbone for trustless, and transparent processes. This architecture ensures that every step of the donation process is traceable, secure, transparent, and tamper-proof, thereby fostering a more efficient and accountable disaster response system.

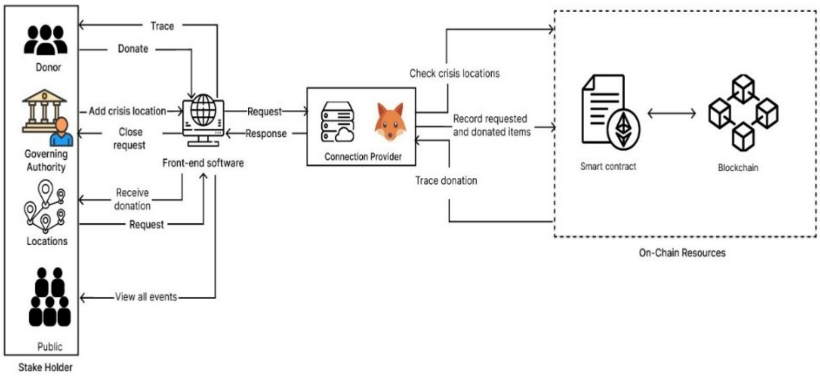

Figure 1 :A high-level architecture for the proposed solution

3.1 Roles of Stakeholder in Proposed Architecture

Each stakeholder plays a crucial role in a proposed architecture. All stakeholders collaborate with each other through smart contract. Figure *2* shows the interaction of stakeholder with smart contract.

3.1.1. Donor

Donors play a crucial role in providing support and resources to those in need. A donor can be anyone who wishes to contribute, whether it's individuals, organizations, or even corporations. In the proposed system, donors interact with smart contracts to record their donations securely. These smart contracts serve as immutable ledgers, documenting the details of each donation transaction, including the donor's accounts, the donated items and quantity. Through systems, donors can also track their donations over time. This transparency and accountability foster trust among stakeholders and enhance the effectiveness of disaster response efforts [20].

3.1.2. Governing Authority

The governing authority plays a pivotal role in overseeing and managing the operations within a proposed disaster donation response system. One of their primary responsibilities is to facilitate the integration of crisis location accounts into the system. When a crisis location requests assistance, the governing authority is tasked with adding the corresponding account address to the system. This action enables the added crisis location to request the resources and prevents the unauthorized to request donations. Additionally, the governing authority is responsible for maintaining the integrity and efficiency of the system by managing the removal of requests that have been fulfilled or are no longer relevant. Once the needs of a crisis location have been met, the governing authority ensures that the corresponding request is promptly removed from the system, allowing for streamlined and updated information.

3.1.3. Crisis Locations

These are entities situated in disaster-affected areas. Once their accounts are authorized and added to the system by the governing authority, crisis locations can submit requests for specific items required for relief efforts. These requests are logged onto the public blockchain, ensuring a transparent record of needs and distributions. Subsequently, crisis locations receive donations from various donors, which they then efficiently distribute among the affected population.

3.1.4. Public

The public compromise of both affected populations and those not directly impacted by the disaster. As stakeholders with access to the system, they have the ability to view all requests for donations and detailed information about the locations involved in the relief operations. The involvement of the public as stakeholders enhances transparency, accountability, and community involvement, ultimately strengthening the effectiveness of the response efforts and promoting resilience in the face of adversity [21].

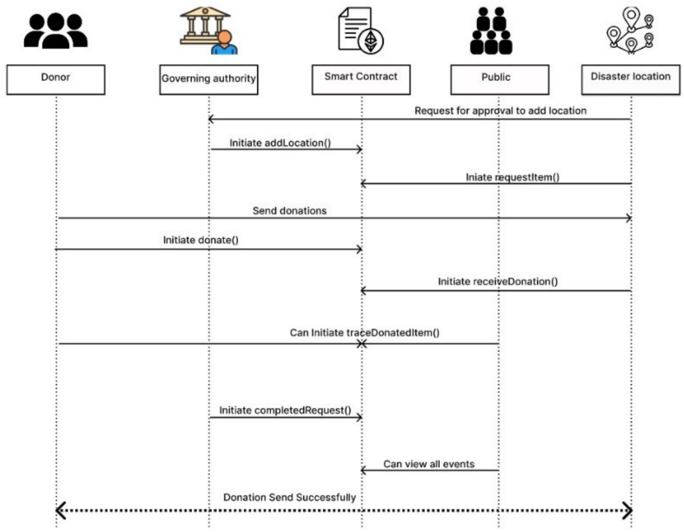

Figure 2: Sequence Diagram showing interaction with smart contract among different stake holders within proposed system

4. IMPLEMENTATION

In this section, developed algorithms are presented along with the implementation detail of the proposed blockchain -based solution for donation management in disaster response. The proposed solution is developed using the Ethereum blockchain platform. The smart contract is written in Solidity language through Remix IDE. Remix IDE is a web-based development environment that allows the user to write, compile, testing, and debugging of the solidity code. The full code has been made publicly available.

The smart contract is deployed and governed by the Government Authority, which serves as its owner. Under this authority, Crisis Locations are established, each identified by a unique Location ID, Name, and Account Address. Crisis Locations have the capability to request essential resources by specifying the required Item names and quantities. Donations are facilitated by Donors, who contribute items along with their respective quantities. Each donation is assigned a unique Donor ID, allowing for precise tracking. Tracing of donations can be initiated by either the Donors themselves or the Crisis Locations. When a Crisis Location has not yet received a donation, its status remains pending. Upon reception of a donation, the status transitions to received. Finally, when all requested items have been successfully distributed, the status is updated to distributed through the completion of the request by the Government Authority.

To provide a detailed understanding of the functions within the smart contract, various algorithms utilized in the proposed solution are outlined. Below are the main functions along with th eir corresponding algorithms:

4.1 AddLocation

Algorithm 1 describes the procedural steps in adding a crisis location. Access to this function is granted to the entity whose account address is the same as the ownerID. The required input to the smart contract needed by function alongside description are presented. After the algorithm is executed, the account address is mapped in accountId ensuring that only authorized entities associated with that registered address can execute the function requestItem.

Algorithm 1: Adding Location

Input: *id: Location ID, loc: Location name, account: Ethereum address associated with*

Output: *Success message if the location is added successfully or Error message if the location*

Steps:

```
// Check if the location ID already exists
if data[id]. itemCount = = 0 then
    require (Location ID already exists);
```

```
    end
    // Assign the location information to the specified location ID
    data[id] = Location (loc, account, new string [ ](0), new uint[ ](0), 0);
    // Update the account ID mapping
    accountId[account] = id;
```

4.2 RequestItem

Algorithm 2 explains the steps to request donation items. This algorithm executes only if the address is present in accountId mapping.

Algorithm 2: Requesting Donation from crisis Location

Input: *itemName: Name of the item, qty: Quantity of the item*

Steps:

```
if caller's address == location's stored address then
    location.items.push: Add the itemName string to the items array
    location.quantity.push: Add the qty value to the quantity array
    location.itemCount += 1: Increment the itemCount in the location object
else
    throw unauthorized messsage
end
```

4.3 ViewRequestedItem

Algorithm 3 is about viewing items that are requested by crisis locations. This function can be called by the public and see request of items through location Id.

Algorithm 3: Viewing Requested Item

Input: *locId: Location ID*

Output*: Array of strings representing requested item names, Array of integers representing corresponding quantities*

Steps:

```
Loads the Location object at the provided locId
if location.account is empty then
    return the location doesn't exist.
else
    return location.items, location.quantity
end
```

4.4 DonateItem

Algorithm 4 describes the steps to donate items. This algorithm can be called by the public who donates. After the function is executed the account address of donor with the details are stored.

Algorithm 4: Donation

Input: *items: Array of item names, quantities: Array of corresponding item quantities*

Output: *Emit DonationEvent*

Steps:

```
Validate input arrays
Create donation record
```

```
        donationId = donationCount
        donationCount += 1
        Create a new Donation object in storage
    Store donations details
        newDonation.donor = msg.sender
        newDonation.items = items
        newDonation.quantity = quantities
        newDonation.donated = false
    emit DonationEvent
```

4.5 ReceiveDonation

Algorithm 5 explains steps for receiving donations. After the request from the crisis location, donors send donations to the crisis location. When they receive donation, they execute this function. This function can be called by the address whose account address is present in accountId.

Algorithm 5: Receiving Donation

```
Input: item: Name of the donated item
Output: ReceiveDonationEvents is emitted
Steps:
Get user ID from sender's address
Access user's location data
for uint i = 0 to ldata.itemCount do
    if compare(item, allitems[i]) then
        quantity required = qty[i];
        requested = true;
        index of requested item = i;
        break;
    end
end
for uint i = 0 to donationCount−1 do
    Donation storage d = donations[i];
    string[] memory items = d.items;
    uint[] memory q = d.quantity;
    for uint j = 0 to items.length −1 do
        if compare(item, items[j]) then
            if q[j] ≥ quantity required then
                q[j]− = quantity required;
                quantity required = 0;
                if q[j] == 0 then
                    break;
                end
            end
            else if q[j] < quantity required then
                q[j] = 0;
                quantity required− = q[j];
            end
```


DOI: doi.org/10.35970/jinita.v6i1.2257

```
            end
            if quantity required == 0 then
                break;
            end
        end
    end
    emit ReceiveDonationEvent
```

4.6 TraceDonatedItem

Algortihm 6 is about tracing donations. The inputs to the smart contract needed and output by the functions are shown with their descriptions. This function can be called by the public for tracing donations of item and status of doantion.

Algorithm 6: Trace Donation

Input: *donationId: Unique identifier for the donation, item: Name of the specific item to track*

Output: *donor: Address of the donation sender, quantityDonated: Initial quantity of the item offered in the donation, receiver: Address of the intended recipient,status: Current status of the*

Steps:

```
Donation storage donation = donations[donationId];
require(donation.donor= address(0), "Invalid donation ID");
string[] memory items = donation.items;
uint[] memory quantities = donation.quantity;
require(items.length > 0, "Donation is empty");
bool itemFound = false;
uint index;
for uint i = 0 to items.length do
    if compare(item, items[i]) then
        itemFound = true;
        index = i;
        break;
    end
end
require(itemFound, "Item not found in the donation");
uint receiverLocationId = accountId[msg.sender];
Location storage receiverLocation = data[receiverLocationId];
require(receiverLocation.account= address(0), "Receiver location does not exist");
DonationStatus donationStatus;
if quantities[index] > 0 then
    donationStatus = DonationStatus.Pending;
end
else if quantities[index] == 0 and =! donation.donated then
    donationStatus = DonationStatus.Received;
end
else
    donationStatus = DonationStatus.Distributed;
end
```


DOI: doi.org/10.35970/jinita.v6i1.2257

return (donation.donor, quantities[index], receiverLocation.account, donationStatus);

4.7 CompletedRequest

Algorithm 7 explains steps after the request from crisis location are fulfilled. This function executes only if it is called by the owner which is governing authority in our solution.

Algorithm 7: Completion of Request

Input: *locId: Unique identifier for the location*

Output*: DistributeEvent is emitted*

Steps:

require*(location.account= address(0), "Location does not exist");*

delete *location.items;*

delete *location.quantity;*

location.itemCount = 0;

emit *DistributeEvent(locId, location.account, location.items, location.quantity);*

5. TESTING AND VALIDATION

This section describes the testing and validation of the developed smart contract's functions. We deploy the smart contract on the Sepolia testnet using the Remix IDE. The deployed smart contract address is 0x678856638fec0fb88fb8ae13bbf310b006975d9c. Table 1 shows the stakeholder addresses within the smart contract that are used during testing. The input data used are assumptions for testing purposes and do not represent real data. The transactions and logs associated with the smart contract's functions are presented below.

Table 1: Account address of stake holder used for testing

Stakeholder	Account Adrdress
Authority	0xe1E6F9C28d604A327DDeEDd15D26D4aa6Ede4630
Locations	0xC24E56A685Ba68bda72132f04368a05E4Ff62e90
Donor	0xbb23067A0a774cA1aA388B46C326eE44182536e5
Public	0xce660cab6d0aaDf91f663B41b4E1A26E034eA448

5.1. Adding Location

This function tests whether the smart contract's owner, acting as the governing authority, can add locations with details such as location ID, location name, and account address. Figure 3 shows the successful execution of the function.

Figure 3: Successful execution of addLocation Function

5.2. Requesting Donation

This function tests whether a crisis location that has been added can make a request with details of items and quantity. Figure 4 shows a successful execution of the function.

Figure 4: Successful execution of requestItem Function

5.3. Viewing Requested Item

This function shows the requested item by entering location id. A successful call of this function is shown in Figure *5*.

Figure 5: Successful execution of viewRequestedItem Function

5.4. Donation

The donateItem function is a key function in this contract. Donors can call this function to provide details about their item and quantity donation. Figure 6 shows a successful execution of the function, along with its corresponding logs and events.

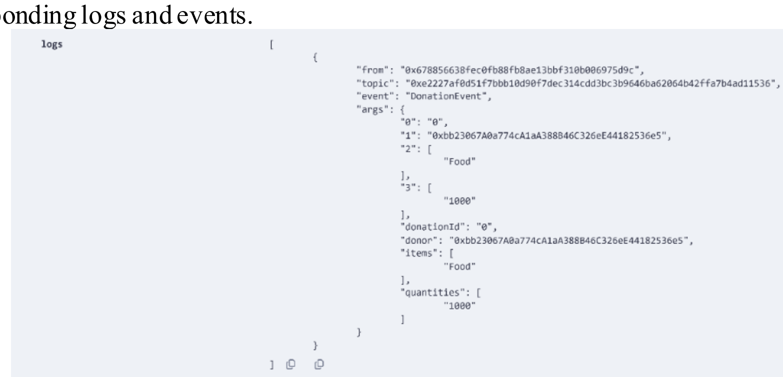

Figure 6: Successful execution of donateItem Function

5.5. Receiving Donation

The receiveDonation function is called after a donation has been received by the crisis locations. A successful execution of the function, along with its corresponding logs and events, is shown in Figure 7.

```
logs    [
            {
                "from": "0x678856638fec0fb88fb8ae13bbf310b006975d9c",
                "topic": "0xd1452972a42ceeaba620033a09e37888294aab1944444ceb48e7a43595eaf45b",
                "event": "ReceiveDonationEvent",
                "args": {
                    "0": "0",
                    "1": "0xC24E56A685Ba68bda72132f04368a05E4Ff62e90",
                    "2": [
                        "Food"
                    ],
                    "3": [
                        "1000"
                    ],
                    "donationId": "0",
                    "receiver": "0xC24E56A685Ba68bda72132f04368a05E4Ff62e90",
                    "items": [
                        "Food"
                    ],
                    "quantities": [
                        "1000"
                    ]
                }
            }
        ]
```


Figure 7: Successful execution of receiveDonation Function

5.6. Trace Donation

This function is called to trace the donation and its status which is pending, received, and distributed. Status 0 represents pending, status 1 represents received and status 2 represents distributed. The successful call of this function is shown in Figure 8.


```
from            0xC24E56A685Ba68bda72132f04368a05E4Ff62e90

to              disasterResponse.traceDonatedItem(uint256,string) 0x678856638fec0Fb88FB8Ae13bBf310B006975D9c

input           0x89d...00000

decoded input   {
                    "uint256 donationId": "0",
                    "string item": "Food"
                }

decoded output  {
                    "0": "address: donor 0xbb23067A0a774cA1aA388B46C326eE44182536e5",
                    "1": "uint256: quantityDonated 1000",
                    "2": "address: receiver 0xC24E56A685Ba68bda72132f04368a05E4Ff62e90",
                    "3": "uint8: status 0"
                }

logs            []
```


Figure 8: Successful execution of traceDonation Function

5.7. Completion of Request

This is the final step in this process, after all the requests have been completed owner executes this function. Figure *9* shows execution of this function with its events and logs.


```
logs    [
            {
                "from": "0x678856638fec0fb88fb8ae13bbf310b006975d9c",
                "topic": "0xaece56e64bbeb191381d8e5c9bd6aacc82f8ce842e9f40833a2593792bc5630a",
                "event": "DistributeEvent",
                "args": {
                    "0": "111",
                    "1": "0xC24E56A685Ba68bda72132f04368a05E4Ff62e90",
                    "2": [],
                    "3": [],
                    "locId": "111",
                    "distributor": "0xC24E56A685Ba68bda72132f04368a05E4Ff62e90",
                    "items": [],
                    "quantities": []
                }
            }
        ]

>
```


Figure 9: Successful execution of completedRequest Function

6. DISCUSSION AND EVALUATION

In this section, the analysis of costs, security in blockchain based donation management and comparison with existing solutions are presented.

6.1. Cost Analysis

In this section, the costs of using Ethereum's smart contracts and making function calls are presented. When you do something on the Ethereum blockchain, like sending a transaction, you have to pay for it called "gas." Execution and Transaction gas are different types of gas fee involved, while running the functions in the Ethereum blockchain.

To analyze the cost of proposed smart contract, Remix IDE is used to estimate the gas cost and deployed on Sepolia Test Network. As of 13 February ,2024 the average gas fee was estimated to 35 gwei and 1 ETH =2669.57 USD according to the etherscan [22]. The cost of deploying smart contract is 173,614,035 gas which is $463.47. Table *2* shows gas fee of the different functions along with the cost in US Dollar to make it easier to understand.

Table 2: Cost Analysis

Functions Task	Execution Gas Fee	Transcation Gas Fee	Cost in USD
Adding Location	101362	100381	18.85
Request Donation	139947	138813	26.05
Doante	169032	166543	31.35
Receive	57540	56732	10.68
Completed Request	55536	44360	9.33

The deployment cost of smart contract is a bit higher, mainly due to the complex structure of the code and the presence of 151 source lines of code. The table indicates that the cost of calling functions is about average because these functions need to record transactions and storage. However, functions like viewRequestedItem and traceDonation don't incur any cost because they only involve viewing data without the need to record a transaction.

6.2. Security analysis for proposed solution

In this subsection, the security analysis of the proposed blockchain based solution for disaster response is discussed where integrity, authorization, availability, accountability are considered as main security goals. It also includes how proposed solutions is resilient against Dos/Ddos attack.

6.2.1. Integrity

The proposed blockchain solution for donation management in disaster response focuses on ensuring traceability and integrity throughout the donation process. Transactions, such as donations and received, are recorded, and stored on an immutable blockchain ledger. This approach allows for transparent tracking of donations enabling stakeholders to monitor the entire process. Additionally, events such as the donations, receive and completed are recorded as immutable entries, providing a clear record of actions taken. This ensures that donations are effectively utilized and allocated to those in urgent need during disaster response efforts.

6.2.2. Authorization

In the blockchain-based donation management system designed for disaster response, the critical functions within smart contracts are restricted to authorized participants using modifiers. This security measure ensures protection against unauthorized access and prevents unwanted entities from utilizing important functions. Such control is vital to ensure that only verified entities are involved in crucial activities such as adding location, requesting resources.

6.2.3. Availability

Blockchain technology is inherently decentralized and distributed, ensuring that once a smart contract is deployed on the blockchain, all logs and transactions become accessible to all participants. Unlike centralized approaches, where transaction data is stored in a single location, in blockchain networks data is replicated across all participating nodes. This redundancy means that the loss of a single node does not result in the loss of transaction data, enhancing the resilience of the system.

6.2.4. Accountability

The traceability of function execution is paramount for ensuring accountability among participants. Just as demonstrated in Section 4 where each function execution in Ethereum records the caller's address on the blockchain. Consequently, all participants are held accountable for their actions. Crisis locations are responsible for requesting donations for which they will be accountable for their decisions and actions. Functions such as "donateItem" or "receiveDonations" will capture the details of who initiated the action, ensuring transparency and accountability throughout the donation process.

6.2.5. Dos/DdoS

The decentralized architecture of blockchains provides resistance against denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks. Additionally, the cost associated with transactions acts as a deterrent for attackers as multiple transactions can become expensive. Blockchain platforms like Ethereum impose transaction fees based on the size of transaction packets, further discouraging malicious activities. By thoroughly understanding potential vulnerabilities and strategically implementing security measures, such as transaction fees and decentralized architecture, a robust and resilient security framework is established for the proposed technique.

6.3. Smart contract security analysis using Slither

Slither is a static analysis framework tool designed for smart contracts. A comprehensive examination to identify and address potential vulnerabilities in code prior to deployment is done using slither. Modifications to smart contract code are only possible before deployment on the main network once deployed changes become irreversible. The analysis revealed one medium-level issue related to owner roles of calling function completeRequest that erases requested items, while no other medium or high-level issues were identified. The result generated by slither is shown in Figure *10* .Importantly, this one medium-level issue was determined to have no adverse impact on the efficiency and security of our smart contract. By employing Slither, the potential risks associated with weak coding practices are mitigated ensuring the robustness and integrity of proposed smart contract.

```
Number of low issues: 0
Number of medium issues: 1
Number of high issues: 0


+-----------------+-------------+------+-----------+--------------+----------+
| Name            | # functions | ERCS | ERC20 info | Complex code | Features |
+-----------------+-------------+------+-----------+--------------+----------+
| disasterResponse |         10 |      |           |          Yes |          |
+-----------------+-------------+------+-----------+--------------+----------+
INFO:Slither:disas.sol analyzed (1 contracts)
```

Figure 10: Security analysis using Slither

6.4. Comparison with existing Blockchain and non-Blockchain solution

Table 3 presents a comparison between proposed blockchain-based solution and existing non-blockchain-based solutions, focusing on critical parameters such as decentralization, transparency, security, traceability, and integrity. The proposed solution stands out as the only decentralized one, offering resilience against single points of failure. In contrast, centralized management in other solutions makes them vulnerable to attacks. Moreover, proposed solutions ensure integrity, traceability, and accountability by storing all actions permanently on an immutable ledger which are not present in other non-blockchain solutions.

Table 3 :Comparison with existing non-Blockchain solution

	i-care [23]	Smart Platform [24]	E-Sharing [25]	Proposed System
Decentralized	No	No	No	Yes
Transparent	Yes	Yes	Yes	Yes
Tracing	No	Yes	No	Yes
Security	No	No	No	Yes
Integrity	No	No	No	Yes

Table *4* shows comparasion of proposed solutions with existing blockchain-based solutions, considering parameters like blockchain platform, mode of operation, cryptocurrency, and their use case category storage. The proposed solution operates on the Ethereum network in a public mode specially categorized for disaster management which distinguished from other existing donation management blockchain solutions.

Table 4: Comparison with existing Blockchain solution

	Proposed System	[13]	[14]	[26]	[27]
Blockchain Platform	Ethereum	Ethereum	Ethereum	Ethereum	Bitcoin
Mode of Operation	Public	Private	Public	Public	Public

Crypto Currency	Ether	Ether	Ether	Ether	Bitcoin
Donation Status	Yes	No	No	No	No
Use Case Category	Disaster Donation Managment	Blood Donation	Covid 19 Pandemics	Disaster aid Crowdfunding	Philantropy donations

7. CONCLUSION & FUTURE WORK

This paper identified critical shortcomings in traditional disaster donation management systems, highlighting issues like centralized control, lack of transparency, and vulnerability to corruption. These limitations often impede effective aid delivery leaving crisis locations communities in dire need. To address these challenges, this paper proposes a novel blockchain-based donation management system. The proposed solution is developed and tested on the Sepolia testnet. The proposed system utilizes smart contracts to ensure transparency, traceability, and security of donations. Detailed cost and security analyses were conducted to evaluate the solution's feasibility and robustness.

While the proposed system presents a promising approach, future work holds the potential to further enhance its impact:

- Automated Donation Allocation: Integrating smart contracts for equitable and needs-based allocation of donations can optimize resource distribution and ensure aid reaches the most critical areas.
- IoT Integration: Embedding real-time tracking capabilities through Internet-of-Things (IoT) devices can enhance supply chain visibility and improve disaster response coordination.
- Real-World Deployment: Testing the system in a real-world disaster scenario will offer valuable insights into its functionality and scalability, paving the way for wider adoption.

By actively exploring these avenues, this research contributes to building a more efficient, transparent, and accountable donation management in disaster response ecosystem, ultimately ensuring that every act of generosity translates into tangible and timely assistance during times of crisis.

REFERENCES

[1] "Global Report on Internal Displacement 2023 IDMC Global Report on Internal Displacement 2023 IDMC II III." Accessed: Feb. 24, 2024.
[2] "INTERNATIONAL DAY FOR DISASTER RISK REDUCTION 2023." Accessed: Feb. 24, 2024.
[3] A. Rawat et al., "The contribution of community health systems to resilience: Case study of the response to the 2015 earthquake in Nepal," J Glob Health, vol. 13, 2023, doi: 10.7189/JOGH.13.04048.
[4] "Digital Accountability," 2023. Accessed: Feb. 14, 2024.
[5] U. S. Mishra, P. Mishra, S. Ranjan, D. ≠1, U. Sankar, and M. ≠2, "Emerging issues and opportunities in disaster response supply chain management," 2013.
[6] S. Park, S. Lee, and J. Cho, "Uneven Use of Remote Work to Prevent the Spread of COVID-19 in South Korea's Stratified Labor Market," Front Public Health, vol. 9, Oct. 2021, doi: 10.3389/fpubh.2021.726885.
[7] M. Fiore et al., "Blockchain for the Healthcare Supply Chain: A Systematic Literature Review," Applied Sciences (Switzerland), vol. 13, no. 2. MDPI, Jan. 01, 2023. doi: 10.3390/app13020686.
[8] Simon, S. Hutagulung, H. Indrajat, and D. Hermawan, "Socioeconomic Factor, Disaster Literacy, and Disaster Awareness in Lampung Province", doi: 10.29313/mimbar.v39i1.
[9] F. Sabz Ali Pour, P. Niculescu-Mizil Gheorghe, and P. Niculescu-mizil Gheorghe, "A Blockchain-Enabled Model to Enhance Disaster Aids Network A Blockchain-Enabled Model to Enhance Disaster Aids Network Resilience Resilience A Blockchain-Enabled Model to Enhance Disaster Aids Network Resilience."
[10] M. Faherza, H. Rahman, I. Nurhadi, S. A. Widjaja, and H. Okitasari, "Blockchain Disaster Management for Humanitarian Logistic as Post-Disaster Response Measures," 2023.
[11] H. Saleh, S. Avdoshin, and A. Dzhonov, "Platform for tracking donations of charitable foundations based on blockchain technology," in Proceedings - 2019 Actual Problems of Systems and Software Engineering, APSSE 2019, Institute of Electrical and Electronics Engineers Inc., Nov. 2019, pp. 182–187. doi: 10.1109/APSSE47353.2019.00031.
[12] A. Almaghrabi and A. Alhogail, "Blockchain-based donations traceability framework," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 10, pp. 9442–9454, Nov. 2022, doi: 10.1016/j.jksuci.2022.09.021.
[13] D. Hawashin et al., "Blockchain-based management of blood donation," IEEE Access, vol. 9, pp. 163016–163032, 2021, doi: 10.1109/ACCESS.2021.3133953.
[14] H. Wu and X. Zhu, "Developing a Reliable Service System of Charity Donation during the Covid-19 Outbreak," IEEE Access, vol. 8, pp. 154848–154860, 2020, doi: 10.1109/ACCESS.2020.3017654.
[15] A. Javadpour, F. S. AliPour, A. K. Sangaiah, W. Zhang, F. Ja'far, and A. Singh, "An IoE blockchain-based network knowledge management model for resilient disaster frameworks," Journal of Innovation and Knowledge, vol. 8, no. 3, Jul. 2023, doi: 10.1016/j.jik.2023.100400.
[16] A. Musamih et al., "A blockchain-based approach for drug traceability in healthcare supply chain," IEEE Access, vol. 9, pp. 9728–9743, 2021, doi: 10.1109/ACCESS.2021.3049920.

[17] D. Kolhatkar, O. Jain, S. Patil, P. Shelke, R. Mirajkar, and P. Wawage, “Blockchain Applications in Disaster Management Systems,” in 2023 4th International Conference for Emerging Technology, INCET 2023, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/INCET57972.2023.10170452.
[18] J. Bhatelia, S. Pandita, A. Ajmera, N. Shekokar, and N. Katre, “Supply Chain of Relief Materials using Blockchain,” ITM Web of Conferences, vol. 32, p. 03036, 2020, doi: 10.1051/itmconf/20203203036.
[19] F. Sunmola and P. Burgess, “Transparency by Design for Blockchain-Based Supply Chains,” in Procedia Computer Science, Elsevier B.V., 2022, pp. 1256–1265. doi: 10.1016/j.procs.2022.12.324.
[20] R. C. Bonfanti, B. Oberti, E. Ravazzoli, A. Rinaldi, S. Ruggieri, and A. Schimmenti, “The Role of Trust in Disaster Risk Reduction: A Critical Review,” Int J Environ Res Public Health, vol. 21, no. 1, p. 29, Dec. 2023, doi: 10.3390/ijerph21010029.
[21] Z. Nkombi and G. J. Wentink, “The role of public participation in disaster risk reduction initiatives: The case of Katlehong township,” Jamba: Journal of Disaster Risk Studies, vol. 14, no. 1, 2022, doi: 10.4102/jamba.v14i1.1203.
[22] “Ethereum Gas Tracker — Etherscan — etherscan.io.” Accessed: Feb. 13, 2024. [Online]. Available: https://etherscan.io/gastracker
[23] N. Izzah Batrisyia Mohd Azhar, M. Mat Rejab, and F. Sains Komputer dan Teknologi Maklumat, “i-Care: A Web-Based Online Goods Donation Management System,” Applied Information Technology And Computer Science, vol. 4, no. 2, pp. 1617–1634, 2023, doi: 10.30880/aitcs.2023.04.02.091.
[24] S. R.Shelar, S. Raju Salve, and A. Santosh Kedari, “A Smart Platform for Donation Handling,” International Journal of Research in Engineering, Science and Management, vol. Volume-3, no. Issue-5, 2020.
[25] H. I. Alzahrani et al., “E-Sharing : Developing a Web Based Online Donation System,” Int J Sci Res Sci Technol, pp. 237–248, Jun. 2020, doi: 10.32628/ijsrst207334.
[26] R. M. Nor, H. Rahman, T. Rahman, and A. Abdullah, “BLOCKCHAIN SADAQA MECHANISM FOR DISASTER AID CROWD FUNDING,” 2017. [Online]. Available: http://www.uum.edu.my
[27] D. Jayasinghe, S. Cobourne, K. Markantonakis, R. N. Akram, and K. Mayes, “Philanthropy on the blockchain,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Verlag, 2018, pp. 25–38. doi: 10.1007/978-3-319-93524-9_2.