LOCATION-BASED SMART CONTRACTS FOR IN-KIND AID DELIVERY

AYNİ YARDIM DAĞITIMINDA KONUMA DAYALI AKILLI KONTRATLAR

AHMET ALPKAN GİRGİN

ASST. PROF. DR MURAT DURMAZ

Supervisor

Submitted to

Graduate School of Science and Engineering of Hacettepe University

as a Partial Fulfillment to the Requirements

for the Award of the Degree of Master of Science

in Geomatics Engineering

September 2024

ABSTRACT

LOCATION-BASED SMART CONTRACT FOR IN-KIND AID DELIVERY

Ahmet Alpkan GİRGİN

Master of Science, Department of Geomatics Engineering

Supervisor: Asst. Prof. Dr. Murat DURMAZ

September 2024, 88 pages

Aid delivery in migration management, disaster recovery and also in emergency situations is a critical process where donors are willing to help those who are in need. Especially in emergency situations the in-kind aid delivery is the first action for many donors who want to directly help immediately to the people in affected region. Most of the time the collection and delivery of in-kind aid is carried out by charitable organizations. However, this raises questions about trust and transparency which may be the main blocking factor in increasing the number of donors. Blockchain technology may provide a means to fill this gap by its distributed ledger and public transactions. In addition, geospatial information technology, especially location-based services can be utilized to improve the transparency of aid delivery. Using location information in smart contracts may improve the transaction of in-kind aid delivery to its intended target location. In this thesis a location-based smart contract concept for in-kind aid delivery is developed and prototyped. The scenarios of actors and roles of organizations in the process of in-kind delivery is listed with a proof of concept smart contract implementation. Furthermore, future research directions and improvements opportunities are listed especially for proof of location concepts.

Keywords: Blockchain, smart contract, location-based services, in-kind aid delivery.

ÖZET

AYNİ YARDIM DAĞITIMINDA KONUMA DAYALI AKILLI KONTRATLAR

Ahmet Alpkan GİRGİN

Yüksek Lisans, Geomatik Mühendisliği Bölümü

Tez Danışmanı: Dr. Öğr. Üyesi Murat DURMAZ

Eylül 2024, 88 sayfa

Göç yönetimi, afet kurtarma ve acil durumlarda yardım teslimi, bağışçıların ihtiyacı olanlara yardım etmek istediği kritik bir süreçtir. Özellikle acil durumlarda ayni yardım teslimi, etkilenen bölgedeki insanlara doğrudan ve hemen yardım etmek isteyen birçok bağışçı için ilk eylemdir. Çoğu zaman ayni yardımın toplanması ve teslimi hayır kurumları tarafından gerçekleştirilir. Ancak bu, bağışçı sayısının artırılmasındaki ana engel olabilecek güven ve şeffaflık konularında soru işaretleri doğurur. Blockchain teknolojisi, dağıtık defteri ve halka açık işlemleri sayesinde bu boşluğu doldurabilir. Ayrıca, coğrafi bilgi teknolojisi, özellikle konum tabanlı hizmetler, yardım tesliminin şeffaflığını artırmak için kullanılabilir. Akıllı sözleşmelerde konum bilgilerinin kullanılması, ayni yardım tesliminin hedeflenen konuma ulaşmasını iyileştirebilir. Bu tezde, ayni yardım teslimi için konum tabanlı akıllı sözleşme konsepti geliştirilmiş ve prototiplenmiştir. Ayni teslim sürecinde yer alan aktörlerin senaryoları ve organizasyonların rolleri, bir kavram kanıtı akıllı sözleşme uygulaması ile listelenmiştir. Ayrıca, konum kanıtı kavramının kullanımı için gelecekteki araştırma yönleri ve iyileştirme fırsatları listelenmiştir.

Anahtar Kelimeler: Blockchain, akıllı sözleşme, konum tabanlı hizmetler, ayni yardım teslimi.

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor Asst. Prof. Dr. Murat DURMAZ for his valuable support and guidance throughout this study.

I am also very thankful to the jury members Prof. Dr. Sultan KOCAMAN GÖKÇEOĞLU, Prof. Dr. Ali Özgün OK, Prof. Dr. Çetin MEKİK, Assoc. Prof. Dr. Saygın ABDİKAN, Assoc. Prof. Dr. Fatih NAR for their valuable comments on the thesis.

I would like to express my gratitude to my esteemed elders, Gökhan and Mehmet, whom I met by chance in the middle of my undergraduate life and who made great contributions to my professional and academic career.

I owe gratitude to my mother Saime, my father Mustafa, and my sister Beyza for their support, patience, and understanding throughout this thesis process, as in every period of my life.

I would like to thank Merve, who is by my side, supports me, and whose love I always felt.

CONTENTS

FIGURES

TABLES

SYMBOLS AND ABBREVIATIONS

Abbreviations

ACTED	Agency for Technical Cooperation and Development
ADM1	1st-Order Administrative Units
ADM2	2nd-Order Administrative Units
ADM3	3rd-Order Administrative Units
AFAD	Afet ve Acil Durum Yönetimi Başkanlığı (Disaster and Emergency Management Presidency of Türkiye)
ASAM	Association for Solidarity with Asylum Seekers and Migrants
AWS	Amazon Web Services
BB	Building Blocks
BFT	Byzantine Fault Tolerance
BLE	Bluetooth Low Energy
CA	Certificate Authority
CARE	Cooperative for Assistance and Relief Everywhere
CRUD	Create Read Update Delete
DFT	Türkiye Development Foundation
DRC	Danish Refugee Council
DTM	Displacement Tracking Matrix
ETH	Ethereum
EU	European Union
GDPR	General Data Protection Regulation
GIS	Geographic Information System
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System

HRDF	Human Resource Development Foundation
HSM	Hardware Security Module
IBM	International Business Machines
ID	Identification
İHH	İnsan Hak ve Hürriyetleri ve İnsani Yardım Vakfı (the Foundation for Human Rights and Freedoms and Humanitarian Relief)
ILO	International Labour Organization
INGO	International Non-Governmental Organization
IOM	International Organization for Migration
KASDER	Kas Hastalıkları Derneği (Turkish Muscle Diseases Association)
KVKK	Kişisel Verileri Koruma Kurumu (Law on the Protection of Personal Data)
LÖSEV	Lösemili Çocuklar Sağlık ve Eğitim Vakfı (Foundation for Children with Leukemia)
MSP	Membership Service Provider
MUDEM	Mülteci Destek Derneği (Refugee Support Association)
NASA	National Aeronautics and Space Administration
NGO	Non-Governmental Organization
pBFT	Practical Byzantine Fault Tolerance
RSA	Rivest–Shamir–Adleman
SAP	Systemanalyse Programmentwicklung (System Analysis Program Development)
SMI	Solution and Mobility Index
TEMA	Türkiye Erozyonla Mücadele Ağaçlandırma ve Doğal Varlıkları Koruma Vakfı (Turkish Foundation for Combating Soil Erosion, Reforestation and the Protection of Natural Habitats)
THK	Türk Hava Kurumu (Turkish Aeronautical Association)

UAH	Ukrainian Hryvnia
UAV	Unmanned Aerial Vehicle
UN	United Nations
UNDP	United Nations Development Programme
UNFPA	United Nations Population Fund
UNHCR	United Nations High Commissioner for Refugees
UNICEF	United Nations Children's Fund
UNO	United Nations Organization
US	United States
UWB	Ultra-Wideband
WASH	Water, Sanitation, and Hygiene
WFP	World Food Programme
WHH	Welthungerhilfe
WHO	World Health Organization
XYO	XY Oracle

1. INTRODUCTION

According to Charities Aid Foundation 4.2 billion people helped someone they did not know in various ways in 2022. This number corresponds to 72% of the adult human population in the world [1]. There is a significant increase in the percentage when compared to the studies of the previous years. For instance, in 2020, this percentage was 55% [2]. Considering the amount of people, the act of helping is an important behavioral model for societies around the world. The types of helping here are in-kind aid, cash aid, or helping another person in other ways (such as helping by using physical strength, knowledge transfer, awareness-raising activities, etc.) by one's own time [1].

According to World Bank data for the period between 2010 and 2019, in-kind aid reaches more people than cash aid in countries within three income groups (low, lower middle, and upper middle) (Figure 1.1) [3]. In addition, according to studies conducted in Türkiye, the percentage of individuals who prefer to make their donations through an organization is 22%. Although it has increased by 10-12% compared to previous years, it is still a low rate. When individuals are asked why they do not donate through an organization, the rate of those who say they do not trust the organizations in question or do not know the organizations has gradually increased over the years and reached up to 28% while other answers are those who say "I donate irregularly and the amount of aid I provide" is very low [4].

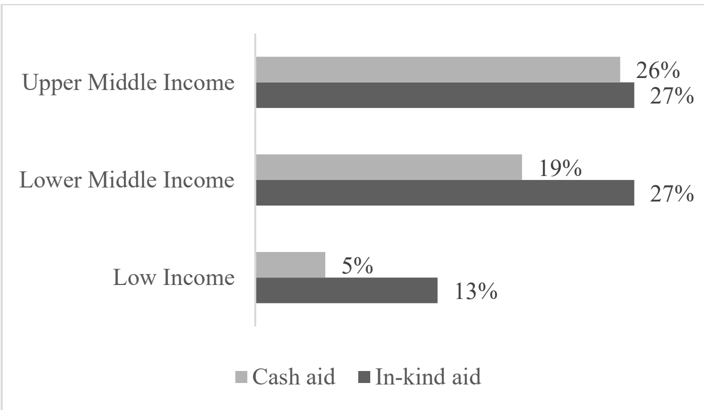


Figure 1.1. Distribution of in-kind aid and cash aid in countries in three income groups (low, lower middle, upper middle) in the period between 2010 – 2019 [3]

Improving trust in organizations requires traceability and transparency which also requires somehow direct contact between a donor and a person in need. In the context of this thesis, an improved process of delivering in-kind aid from donors through organizations is discussed. Transparency and traceability are determined as priorities for the design of the processes and blockchain technology is used in this context. Adjustable transparency in the blockchain is a feature that can make a difference. In addition, since the sensitivity level of various data of some actors is high in the process of delivering in-kind aid, using a blockchain-based approach instead of a traditional database management system may provide benefits in terms of data security and anonymity.

As a result of the research conducted during the preliminary studies, within the scope of this thesis, the most suitable blockchain development environment is the Hyperledger Fabric framework. Hyperledger Fabric provides a good basis for developing solutions with modular architecture. Hyperledger Fabric is a versatile framework designed to meet the needs of many different industries. Additionally, it offers an approach that provides scalable performance while preserving privacy [5]. Considering all the features of Hyperledger Fabric, it has become an unofficial standard for enterprise blockchain platforms [6].

1.1. Problem Definition and Motivation

One of the biggest problems donors face when they request to help those in need through institutions and organizations is the trust problem towards these organizations. The main reason behind the trust problem is that donors do not know whether the in-kind aid they donate reaches those in need. In situations such as disasters where humanitarian aid is an urgent need, the fact that donated in-kind aid is not sent to those in need on time is another factor that creates distrust. Such trust problems can reduce the level of solidarity in society and cause insufficient aid to reach people in need, especially in emergencies.

The general lack of transparency in aid delivery processes is the main cause of this distrust. If donated aid materials are included in a step-by-step traceable system, the trust problem among donors may be eliminated. If the trust of donors is gained and a better perception is created for humanitarian aid organizations, the desire to help at the donor level may potentially increase and people who have never helped before may become donors, furthermore effective aid network can be established in society.

It is also essential that donated in-kind aid is delivered to those in need in an orderly manner. If donated in-kind aid is distributed haphazardly, it may result in irrelevant items reaching people in need. From a different perspective, it may also cause people in need to not receive the items they need. Another negativity is the excessive accumulation of items declared as needs, which restricts access to aid for people in need in other regions. This irregularity not only means a waste of aid resources but also causes a loss of effort for all professionals and volunteers working in the field of humanitarian aid.

The irregular and uncoordinated distribution of in-kind aid is a problem. A system in which there is a central institution for the coordination of the distribution of in-kind aid and where all relevant institutions and organizations record the in-kind aid sent to those in need may prevent this irregularity and lack of coordination. However, centrality shall not shadow transparency and autonomy.

Blockchain technology and distributed ledger may provide a solution to the problem. The use of geospatial information especially location information within blockchain transactions may also improve traceability and transparency. The main goal of the thesis study is to research the potential use of location information and location-based smart contracts in the domain of in-kind aid delivery in migration management, disaster recovery, and emergency. Location-based in-kind aid delivery within a blockchain framework may provide instantaneous in-kind aid delivery status for both donor and people in need while preserving privacy and improving coordination among organizations. To achieve the goal the following objectives are established.

- The current state of the use of location information in Blockchain technology for in-kind aid delivery,
- The analysis and design of a process leveraging Blockchain technology for in-kind aid delivery.
- Concept development of location-based smart contracts to support transactions in the in-kind aid delivery.

1.2. Structure of the Thesis

The manuscript is structured in the following chapters. The first chapter provides an introduction to the problem with motivation, goals, and objectives. The second chapter provides background information on aid delivery, blockchain, and related technologies. The next chapter describes the proposed concept for in-kind aid delivery discussing organizations, roles responsibilities and use cases. The fourth chapter provides the design of blockchain database and proof of concept smart contract implementation with discussions. Finally, the last chapter provides the concluding remarks.

2. BACKGROUND INFORMATION

This chapter provides a brief summary of the background of the study. First, the concept of aid delivery and the main issues concerning the trust of donors and person in need are discussed with references given as necessary. Followed by background information on blockchain technology and concepts related to the technology, the Hyperledger Fabric framework is introduced. The use of blockchain technology in humanitarian assistance is listed with references and the use of geospatial data in blockchain technology is presented. Finally, the use of geospatial data in humanitarian assistance is provided.

2.1. Aid Delivery

As of 2024, nearly 300 million people worldwide are in need of humanitarian assistance and protection due to conflicts, climatic causes, natural disasters, and other factors [7]. There are also many different humanitarian assistance institutions and organizations that provide services on a very wide spectrum in an environment where there are so many people in need. While some of these institutions and organizations provide services on a national scale, there are also those who provide services on an international scale.

Some of these institutions and organizations carry out humanitarian assistance activities by distributing in-kind aid to various disadvantaged groups. In the process of identifying these disadvantaged groups, sometimes the quantitative and qualitative results of a comprehensive and scientific field assessment play a role, while sometimes they can also be shaped by simple interviews with government agencies, local and international non-governmental organizations (NGOs), and local communities. In some cases, there may not be a sufficient budget to perform the field assessment mentioned at first. Even if there is a budget, there may not be the necessary time according to the urgency of the situation. For this, in some cases, it is necessary to proceed with the second-mentioned method. Under ideal conditions, it is expected that these two situations will be harmonized in the process of identifying disadvantaged groups.

Some aid agencies and organizations not only deliver in-kind aid that they purchase with their own budgets, but they also accept donations from donors and deliver them to those in need. Organizations such as LÖSEV (Foundation for Children with Leukemia) [8], TEMA (The Turkish Foundation for Combating Soil Erosion, for Reforestation and the Protection of Natural Habitats) [9], Darüşşafaka Society [10], KASDER (Turkish Muscle Diseases Association) [11], Ahbap Association [12], THK (Turkish Aeronautical

Association) [13], Yeşilay (Turkish Green Crescent Society) [14], Türk Kızılay (Turkish Red Crescent Society) [15], and İHH (Humanitarian Relief Foundation) [16] can be given as examples of organizations with a large impact area that accept in-kind aid from donors in Türkiye. In 2021, the percentage of people making donations through an institution or organization in Türkiye is 22% (Figure 2.1). Although this rate is higher than the studies conducted in previous years, it is not at the desired level yet [4].

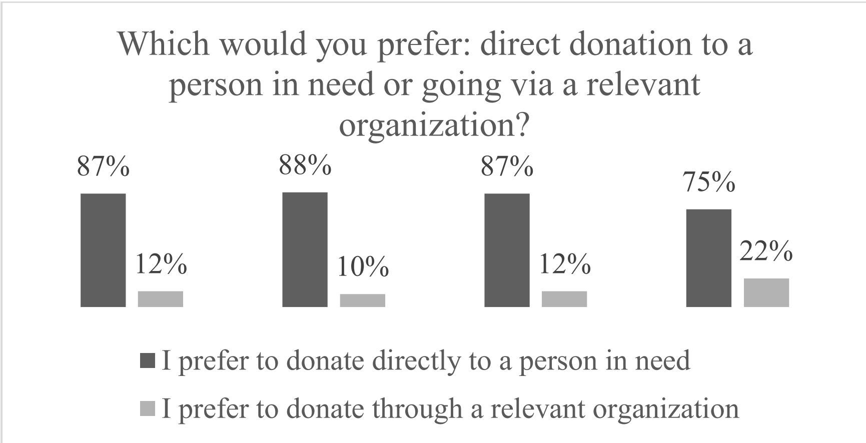


Figure 2.1. Which would you prefer: direct donation to a person in need or going via a relevant organization? [4]

In the same study, the reason why people do not provide aid through organizations was investigated and 21% of the participants answered that they do not trust the institutions in question, and 7% answered that they do not know the institutions in question. The remaining two answers in the majority are reasons related to the individuals themselves. In addition, 45% stated that they provided aid irregularly and 24% stated that the amount of aid they provided was too small. Considering the percentages for organization-oriented reasons, it is very clear that there is a lack of confidence in the institutions and organizations that carry out humanitarian assistance activities. When those who donated to institutions were asked "Why did you choose this institution?", answers confirmed the assertion about distrust in institutions. Accordingly, the first answer among the answers given to this question was "I think this institution is transparent / I trust this institution" with 41% [4].

According to parallel studies conducted in Türkiye and around the world, the percentages of people helping a stranger or a person they do not know have also been compared. The percentage in Türkiye is 34%, while is 59% worldwide. The fact that this ratio is low in Türkiye compared to the world is an interesting and necessary issue to study, and this situation has been examined in detail in the continuation of the research. In the research, the reason for this distrust towards institutions and organizations has also been studied. Accordingly, donors in Türkiye do not receive any detailed reports in return for their donations to institutions and organizations. It has been stated that this situation is the most important factor in the inability to establish a long-term trust relationship with donors. Seventy-three percent of the donors stated that they want the institution or organization to which they donated to maintain communication with them. In addition, donors were also asked about the factors they considered important when choosing the institution to donate to. The most common answer given by the participants, with 38%, was the trust that the institution will use the donations it receives for its intended purpose. According to the research, institutions or organizations do not give any feedback to donors, while donors demand that institutions or organizations establish a long-term relationship with them. Regarding this, another question asked to donors was "What kind of information do they want?" to those who want to continue contacting them, the most preferred type of information is about how the donation is used. Suggestions on what will encourage people to donate more in the next year have also been queried. Among the answers, non-governmental organizations being more transparent and being sure of how donations are spent are among the top three most agreed upon propositions [4].

In general, transparency and accountability are the most important points in the aid activities carried out through institutions and organizations. The two things that donors attach the most importance to when carrying out aid activities through an institution or organization are trust and the use of their donations according to their purpose. Donors naturally do not want their donations to go to waste. From this perspective, the most important factor that will increase donations made through institutions or organizations is to establish a healthy relationship of trust between institutions/organizations and potential donors. An information structure about the use of donations is generally accepted. [4]. Based on all these research results, a platform where donors can examine the status of their donations, track their geographical location, in another sense, track their donation and check whether it has reached those in need will be a platform that will be very welcomed by the donor. This kind of platform requires information systems to be established and used.

2.2. Centralized and Decentralized Systems

In a broad sense Information Systems can be considered into two groups: centralized and decentralized systems. In centralized systems, there is a central structure equipped with administrative powers. Although this structure, where everything is managed from a single source, is easy to design, sustain, trust, and manage, it also has inherent limitations in many aspects. Centralized systems are more vulnerable to attacks, which makes them expensive to secure. The concentration of power in a single place may lead to unethical operations. Additionally, scalability is often not easy for these systems [17].

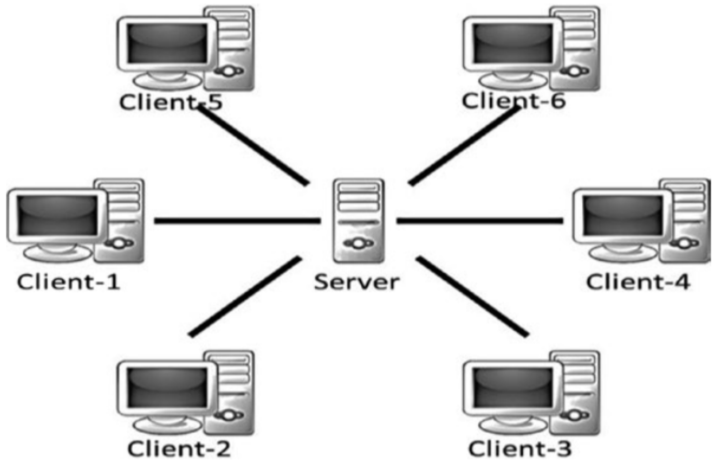


Figure 2.2. An example of a typical centralized system [17]

In decentralized systems, there is no central mechanism with complete control, and each node has equal authority. Since there is no central structure that can be easily targeted and attacked, it is difficult to tamper and therefore more secure. Decentralized systems are inherently more egalitarian in perspective. Unethical operations are less common in these systems [17].

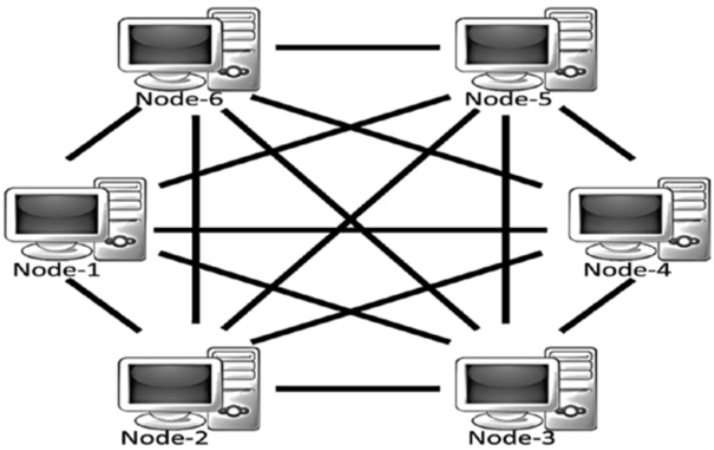


Figure 2.3. A decentralized and peer-to-peer system [17]

2.3. Blockchain Technology

The concept of blockchain was first put forward by Satoshi Nakamoto (assumed to be a nickname, thought to be used by individual or groups [18]) in their article "Bitcoin: A Peer-to-Peer Electronic Cash System" published in 2008 [19]. According to this article, e-commerce is almost dependent on financial institutions that serve as a reliable third party. Here, instead of relying on an independent third party, an electronic payment system based on cryptographic proof is proposed, where transactions are ordered chronologically with a peer-to-peer distributed timestamp server, allowing parties to transact directly with each other. According to the article, the only way to confirm whether a transaction has occurred or not is to be informed of all transactions made. In the traditional model of relying on a trusted third party, the third party is aware of all transactions and decides which should occur first. In order to do this without a trusted third party, transactions must be publicly published [19]. The steps required to run the blockchain network are as follows:

1. Newly executed transactions are announced to all nodes.

2. All nodes collect new transactions into a block.

3. All nodes operate on a proof of work within their block.

4. When a node solves the proof of work, it announces this block to all other nodes.

5. Other nodes accept this block only if all the transactions in it are valid and have not been spent so far.

6. Nodes passively announce their acceptance of the block. When nodes work to create the next block in the chain, they use the hash of this block as the previous hash, thereby indirectly announcing that they have accepted the node [19].

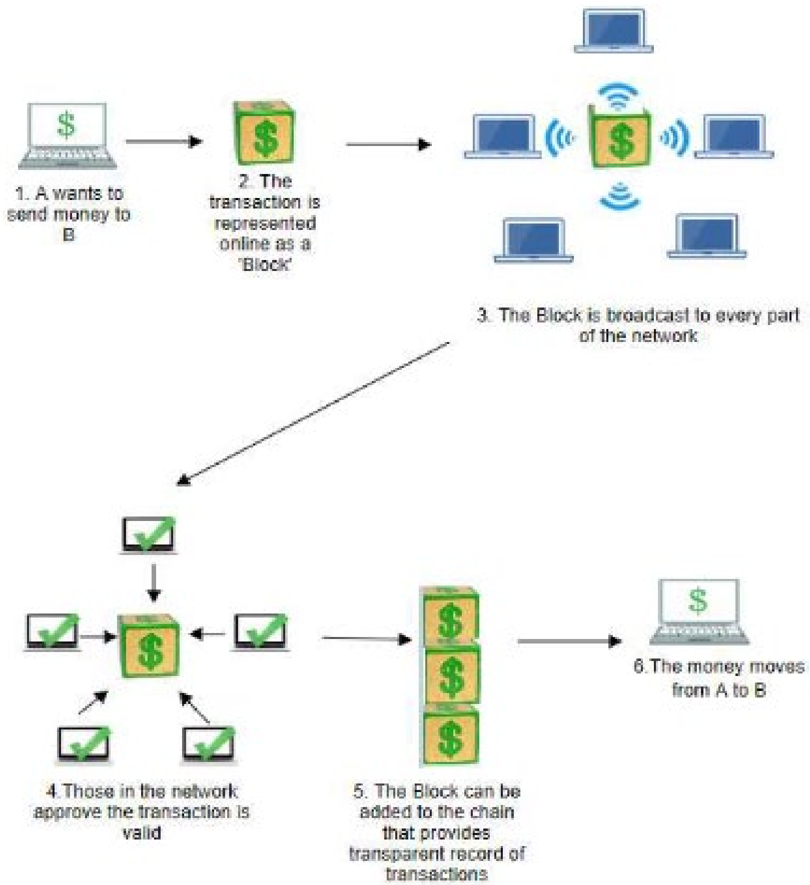


Figure 2.4. A figure showing how blockchain works [20]

Due to its structure, the first transaction in the block of a chain is an exceptional transaction in which the money belonging to the creator of the block is created. These exceptional transactions both encourage nodes to support the blockchain network and enable money to be put into circulation for the first time. Since there is no central authority to print money in systems dominated by the blockchain, the regular circulation of a fixed amount of newly printed money is similar to gold miners spending their resources and releasing gold, a valuable metal, into circulation. In this case, the resources spent are the time of the processor and electricity [19].

In traditional banking, access to information is limited to interested parties and a trusted third party (financial institutions, banks in general), ensuring a certain level of privacy. At first glance, the method of publicly publishing all transactions on the blockchain seems to be contrary to privacy, but privacy is ensured by anonymizing the public keys through which the flow of information is provided. In other words, everyone can see the amounts that the two parties have sent to each other, but they cannot access information about who the subjects of the transaction are [19].

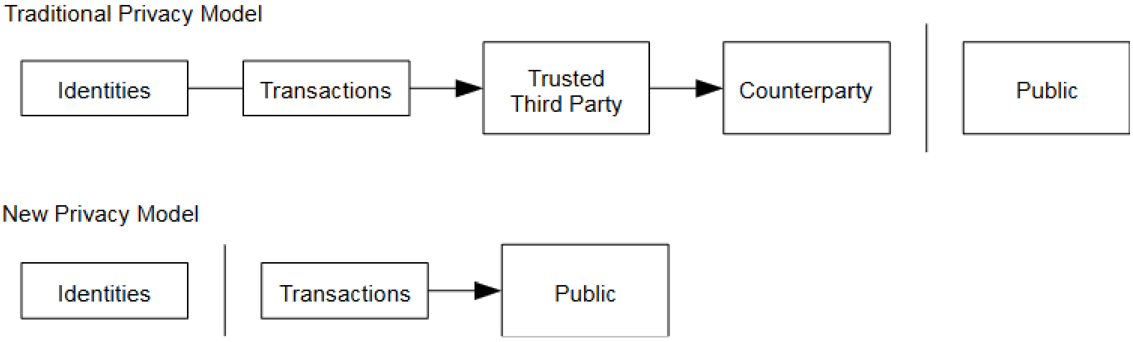


Figure 2.5. Traditional privacy model vs new privacy model [19]

One of the concepts underlying blockchain technology is cryptography. The main purposes of cryptography include ensuring information security [21], identity validation, ensuring confidentiality, ensuring integrity, and non-repudiation. Two methods are used for encryption in blockchain technology. These are “symmetric encryption” and “asymmetric encryption” methods. A comparison of these two techniques is also given in Table 2.1. In symmetric encryption, the same key is used when both encryption and decryption processes are performed [22]. A secret key, which can be a number, a letter, a word, or just a series of random letters or numbers, is applied to the message to change the content. This method can be applied as simply as shifting each letter in the alphabet a few digits, or it can also be applied in a more complex way [23]. The most important advantage of this method is that it has low data processing power and very fast operation. In asymmetric encryption, different keys are used during the encryption and decryption processes. In this method, one set of keys is public while the other is private [22]. It is possible for anyone who has a private key to decrypt the message text, so the private key is kept confidential. In an ideal scenario of usage, only the owner of the key knows what the private key is [23]. In this context, the safekeeping of private keys is the responsibility of the owner. In the cryptocurrency structure, if the owner of the private key loses this key, it will not be possible for this user to verify who owns the encrypted transactions. If this private key is obtained by another ill-intentioned user, it may lead to the ownership of assets linked to this key changing hands. In order to prevent such problems, it seems possible to establish structures similar to intermediary institutions that will protect the keys of users [24].

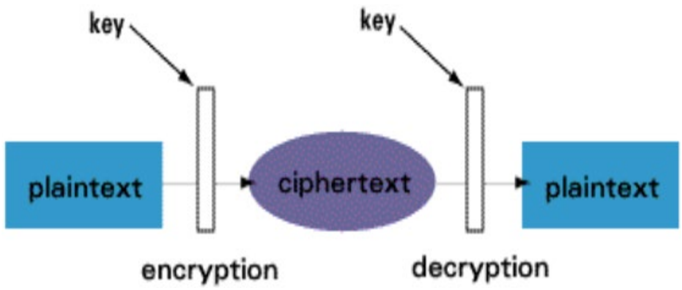


Figure 2.6. Symmetric key cryptographic algorithms process [23]

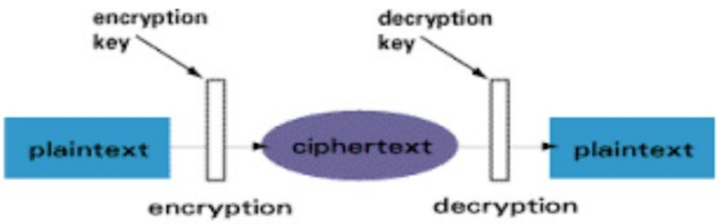


Figure 2.7. Asymmetric key cryptographic algorithms process [23]

Although blockchain technology has made a name for itself with Bitcoin, the origins of this technology date back to old times [25]. In the article titled "How to time-stamp a digital document", which S. Haber and W. S. Stornetta [26] worked on together in 1991, the approach that ensures that digital documents cannot be changed by time-stamp is considered one of the pioneering ideas behind the blockchain. The first example of blockchain technology is Bitcoin. Like most new revolutionary technologies, Bitcoin has not suddenly made the world talk about itself. Since the price of Bitcoin started to rise in 2011, awareness about it has increased exponentially over the years. Over time, the fame of Bitcoin has also become a misdirection for the concept of "blockchain", the blockchain is much more than a cryptocurrency and has the potential to make major impacts [25]. One of the important quotations describing this situation is expressed as "It's not Bitcoin, the still speculative asset, that should interest you" by Don and Alan Tapscott in their work called Blockchain Revolution [27]. Just one of the numerous uses of the blockchain is Bitcoin, as well as cryptocurrencies. The technology that enables the creation of an environment of trust between individuals and institutions through the decentralization of the blockchain was understood relatively later [25].

Table 2.1. Comparison of symmetric encryption and asymmetric encryption [28]

Key Differences	Symmetric Encryption	Asymmetric Encryption
Data size	Utilized for large-scale data transmission.	Utilized for small-scale data transmission.
Resource Utilization	Symmetric key encryption uses few resources when operating.	Asymmetric key encryption uses a high consumption of resources.
Key Lengths	A key size of 128 or 256 bits.	RSA key size of 2048 bits or more.
Security	Less safe since encryption is done using a single key.	Much safer since encryption and decryption require two keys.
Techniques	It is an outdated method.	It is a contemporary encryption method.
Confidentiality	There is a danger that a single key used for encryption and decoding may be compromised.	There is no longer a need to share keys thanks to two distinct keys designed for encryption and decryption.
Speed	Symmetric encryption is a quick method.	Asymmetric encryption is a slow method.

The announcement of a decentralized, open-source, and smart contract-supported blockchain network called Ethereum in 2015 has been an important stage for this technology [25]. Ethereum's application layer consists of smart contracts. Computer programs stored on the blockchain follow the logic of "If this happens, do this" and guarantee that it will work within the rules specified in its code. First, Nick Szabo introduced the concept of a "smart contract" [29]. He has two main articles on this subject: "Smart Contracts" in 1994 [30] and "Smart Contracts: Building Blocks for Digital Markets" in 1996 [31]. Szabo predicted that automated and cryptographically secure transactions and business activities in a digital market can be performed without trusted intermediaries. This prediction was realized by smart contracts on Ethereum [29].

In traditional contracts, there is a need for people who will realize the results of the contract, and this is why traditional contracts are problematic. Even if the terms of the agreement are met, you must trust another person to fulfill the agreement. The signing of traditional contracts does not mean that the contract has come to an end; the realization of obligations arising from the terms of the contracts is also a great burden. The cost, need for trust, time loss, and workload incurred in the process from the writing of traditional contracts to the latest completion are much different compared to smart contracts that are developing today. The biggest benefit of a smart contract is that it runs the code precisely when certain conditions are met. The result does not need to be interpreted or proven. This eliminates the need for reliable intermediaries. Smart contracts are useful for monitoring and auditing. Since the smart contracts on the Ethereum blockchain are in a public structure, anyone can instantly track asset transfers and other relevant information. In addition, in the same way as traditional contracts, their content can also be checked before interacting with a smart contract. Smart contracts also provide privacy. Since transactions on Ethereum are publicly linked to a unique cryptographic address, the privacy of users is also protected from observers [29]. However, it would not be correct to say that this privacy also brings with it absolute security. According to a study, 8,833 of the 19,366 smart contracts on the Ethereum network, that is, 46%, were found to contain potentially documented bugs. The total value of these contracts with bugs is approximately 3.1 million Ether (the value at the time of the study is approximately 30 million US dollars.) [32].

Smart contracts are a kind of computer program; where making calculations, creating currencies, storing data (Figure 2.8.), minting NFTs, sending messages, and designing graphics can be shown as examples of what smart contracts can achieve [29].

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.4.16 <0.9.0;

contract SimpleStorage {
    uint storedData;

    function set(uint x) public {
        storedData = x;
    }

    function get() public view returns (uint) {
        return storedData;
    }
}
```

Figure 2.8. A simple smart contract that features storing and retrieving a uint (unsigned integer) [33]

Although smart contracts are thought to have been used first in the Ethereum blockchain network because they are used extensively in this chain, they can also be used, albeit at a simple level, in the Bitcoin network. In addition, this structure is used on many different blockchain networks, as in Hyperledger Fabric, although sometimes with a different name (chaincode). As a smart contract development environment, each blockchain network has its own style, some networks use existing programming languages (Go, Java, C#, etc.), while some networks use programming languages that they have developed themselves (Solidity and Vyper, both of which are being developed by Ethereum.) [24].

Blockchain is based on establishing trust in a distributed architecture [34]. One of the factors that provides this trust is that the blocks cannot be changed. Each of the nodes in the blockchain has a copy of the blockchain (Figure 2.9.). Blocks consist of two main parts, the heading part of these is connected to the first block of the chain. For this reason, the heading of new blocks must contain the hash code of the previous block, so no one can make changes regarding the previous block. The other part of the block, the body content, contains the verified list of transactions, the type of transaction, the addresses of the parties involved in the transaction, and other relevant details. Thanks to this structure of the blockchain, when looking at the last block in a chain, the non-falsified information of all previous blocks can be accessed [17].

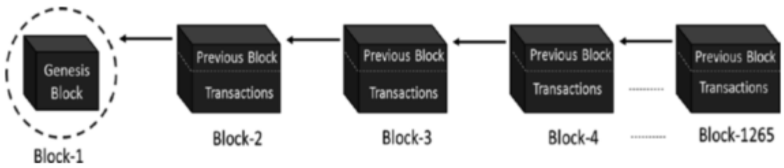


Figure 2.9. The data structure of blockchain [17]

Blockchain technology offers a sophisticated approach to data storage. Unlike traditional centralized servers, blockchain data storage provides a method of dividing data into pieces and distributing them across multiple cloud-based storage devices. Using blockchain for data storage simplifies file transfer across the network without the need to route through a limited number of data centers. However, as blockchain networks expand, data duplication across nodes can lead to escalating storage issues, resulting in challenges with availability, performance, and scalability. Storage remains one of the most frequently mentioned challenges facing blockchain technology nowadays [35].

2.3.1. Consensus Mechanisms

Consensus is needed on a block basis for a healthy chain structure. Nodes should not offer a block to other nodes at the same time. The process of selecting the node to propose the block is a topic of discussion in itself. In this context, there are various proof methods and they are selected in such a way that they are most appropriate according to the structure and needs of the blockchain. Ideally, these blockchains should be designed so that nodes get the most benefit when they perform the right things. One of the most effective ways to ensure this benefit-seeing mechanism is through giving encouraging rewards to the right activities and punishing malicious activities. But there is a problem here: in public blockchains (such as Bitcoin) people have the opportunity to have more than one anonymous public identity. As a result, it becomes difficult to punish such people. The methods followed here are important; punishing people may not work if their public identities are anonymous, but it may work if their public identities are not anonymous. Even if it is considered that there is a very sufficient mechanism to choose the next block in the first place, it may be necessary to choose a different method of proof just because of these reward and punishment situations [17].

Some consensus mechanisms are introduced below:

2.3.1.1. Proof of Work (PoW)

The history of the PoW consensus mechanism dates back to old times [17]. In an article titled "Pricing via Processing or Combatting Junk Mail" written by Moni Naor and Cynthia Dwork in 1993 [36], it deals with some of the work of a user who has requested a service in order to prevent abuses such as spam in e-mails [17]. In 1999, the term "Proof of Work" was used for the first time in the article "Proofs of Work and Bread Pudding Protocols" by Markus Jakobsson and Ari Juels [37]. Despite all this, with the use of it in Bitcoin, it has turned into a more popular concept [17].

The underlying idea of the PoW mechanism is that nodes must perform a certain amount of work within a transaction block before announcing the block to the entire network. PoW is a piece of data that is difficult to produce in terms of calculation processes and time extent but is also easy to verify. According to the spam-preventing function, which is one of the first uses of PoW, a person must do a certain amount of work before sending an email, and if they do not do it, it will be considered spam. Its use in the blockchain is in a similar way. If some calculations are performed and intensive work is performed before a block is generated on the network, this will help in two ways; firstly, this process will take some time, and secondly, if an ill-intentioned node tries to inject an imposter transaction into the block, the other nodes in this block will reject it, putting the node announcing the fraudulent block into a very costly situation. The reason for this is that the calculations made in order to obtain the PoW will have no value, which will only return to the ill-intentioned node as a waste of time and electricity consumption expense. If nodes could do the job of proposing blocks without any effort, this would be an inviting structure for ill-intentioned nodes. In such a structure, nodes proposing an imposter block and its rejection would not pose any trouble for these nodes, since there would be no time to waste or any cost to pose difficulties. On public blockchain networks, users who exhibit honest behavior and do not engage in fraud should be rewarded. If this reward system does not work properly, the profit motivations of users (miners) who invest in high-tech computer parts that can solve these complex problems for PoW will decrease and the system will become unsustainable [17].

The most problematic feature of PoW is that it causes too much energy consumption and therefore carbon footprints. If Bitcoin, the most important blockchain network implementing the PoW protocol, were a "country", it would be 27th in the list of countries that consume the highest amount of energy, which means that Bitcoin surpasses 166 countries in the world in terms of electricity consumption. Bitcoin, which uses the PoW protocol, not only consumes a large amount of electricity but also creates a significant water footprint. According to this, the water footprint of Bitcoin is great enough to meet the current domestic water needs of more than 300 million people in sub-Saharan Africa [38]. In addition, cryptocurrencies with the PoW protocol are being tried to be regulated in the EU. Although there are no restrictions at the moment, various draft rules have been put in place to protect consumers and make mining more sustainable [39].

2.3.1.2. Proof of Stake (PoS)

In the PoS mechanism, the concept of verifier stands out instead of miner. In this mechanism, verifiers are required to guarantee their stake in order to be a candidate for verifying transactions. The greater the amount of stake mortgaged by a verifier, the greater the probability of verifying a new block of transactions. In this context, verifiers are required to prove that they own a certain percentage of all stakes in a particular digital currency within a certain period of time [17]. For instance, in order to be a verifier on the Ethereum network, it is necessary to have a minimum of 32 ETH. Verifiers have the responsibility of storing data, realizing transactions, and adding new blocks to the blockchain. Rewards are also given for the actions taken by the verifiers, who are the gears of the consensus mechanism. Verifiers, who properly separate transactions made on the network into blocks and control the activities of other verifiers, are also rewarded for these activities, which is exactly what keeps the chain secure. While verifiers protect the network in this way, they consume much less energy compared to the PoW system in terms of energy consumption [40]. In PoS mechanisms, although the block creator varies depending on the amount of stake it has, it is generally more decisive, so it has a much faster structure than PoW mechanisms. In addition, PoS mechanisms can provide a more effective security cordon against ill-intentioned activities compared to PoW, because performing a malicious activity means that the entire amount of stake in question is put at risk [17].

2.3.1.3. Practical Byzantine Fault Tolerance (pBFT)

The pBFT is a practical mechanism developed by putting the original Byzantine Fault Tolerance (BFT) on the base in order to protect distributed networks from attacks. In this mechanism, compared to PoW, where the security threshold is 51% if the attacker users do not capture more than 50% of the general resource, it is ensured that an absolutely secure transaction can be performed. In pBFT, attackers must be less than 33% of the total users in order for the mechanism to be immune to attacks [41].

The pBFT mechanism basically consists of four stages (Figure 2.10.). First, the client sends a request to Replica 0 (primary node) to call any service transaction. This request to the primary node is transmitted by the primary node to other nodes. Then the other nodes examine the request and send their responses to the client. Finally, the client waits for answers from different nodes with the same response until f + 1 (f means the number of incorrect nodes in the group) responses are answered, terminates the process when there are as many responses as f +1, and transmits the results to all nodes [41].

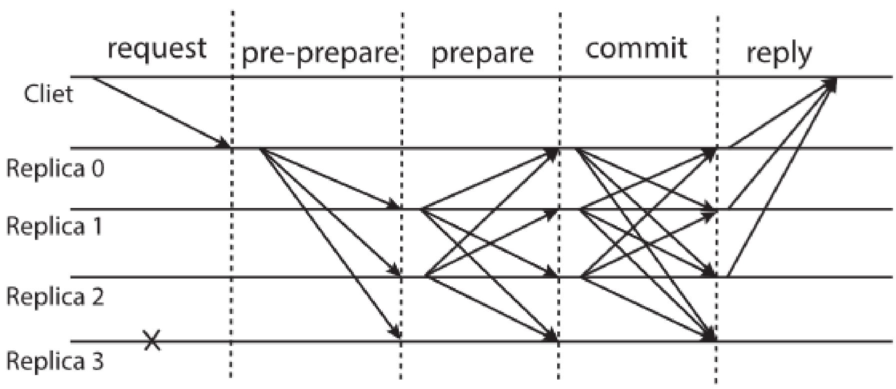


Figure 2.10. pBFT consensus processing [41]

pBFT is a popular mechanism for private and corporate blockchains since it involves less energy consumption and less complexity (and is therefore faster) [41]. In addition, pBFT abstracts ownership such as hardware and stake in a different way from approaches such as PoW and PoS. In other words, if we think of the blockchain as a parliament, just as one member of parliament enables the parliament to make decisions, in blockchain networks with this consensus protocol, the lowest level user also has a say in the operation of the network. In this approach, all verifiers in the network should be aware of each other, and the new verifiers that will be included in the blockchain network should be approved by a centralized structure. In fact, this situation contradicts the decentralization feature of the blockchain. For this reason, this consensus structure is considered more in private networks rather than open networks [24].

2.3.2 Proof of Location (PoL)

PoL is a consensus method that confirms that a living being or an inanimate object is located in a certain geographical location at a certain time [42]. Services that are location-based are becoming more and more crucial. While conventional GPS systems work well for location reporting in general, they have issues with security and indoor tracking. Low-security location reporting systems have several drawbacks, such as the potential for UAVs to be tricked and pose a threat to national security [43].

A method that combines GPS signals with access network channels and cellular tower triangulation using a multi-channel system has been proposed for location verification but ill-intentioned people can bypass such systems. GPS signatures, another recommended verification technique, are not impervious to forgery. In this regard, hardware-based verification approaches have also been developed to determine the proximity of a device. Nevertheless, ill-intentioned people may also readily alter the location data these devices offer through the use of signal attenuation. Additionally, channel noise, deployment complexity, and line-of-sight constraints are issues with these systems [43].

Another solution that is offered for location proofing is the use of sensor readings and the trusted platform module [44]. In addition, in one of the studies conducted for secure location proof, an architecture based on a location authority (a stationery entity that is responsible for providing location proofs and has a unique identifier) and a witness (another mobile user who can claim that the mobile device is at a certain location. The said witness is not an eyewitness but a person who is in a common spatial and temporal location with the user at a certain location.) was developed [43].

PoL's first mention of blockchain technologies took place at Ethereum's DevCon 2 in 2016. Lefteris Karapetsas, an Ethereum developer, used a concept called “Proof of Presence” in their Sikorka project to enable the deployment of smart contracts in the real world [45]. Issues such as supply chain, parametric insurance, identity validation applications, and consumer transactions are among the important areas of location-related implementations. For example, for a user who buys a product from an e-commerce website, it may be necessary to prove by geographical location that the product was delivered to the user so that the price of this product can be automatically transferred to the commercial company from their digital wallet via a smart contract [46]. In a study that addresses a similar issue to this example, it is very important for sellers, buyers, and logistics service providers that the location is reliable and publicly verifiable [47]. In this study, there is a witness and a prover, similar to the previous study (Khan et al.) [43]. In this study, the concept can be simply explained as the witness node sends the location proof to the prover node and the prover node verifies this location. This concept has three main steps [47].

1. The location proof request is sent to the witness node by the prover node. While the witness node proves this request; the geographic location must be within short-range communication range, the request in question must refer to the last block in the chain, and must be signed with the private key of the proving node [47].

$$Request_{a \to b} : \begin{bmatrix} PublicKey_a \\ (Lattitude, Longitude)_a \\ h(Block_{x-1}) \\ timestamp \end{bmatrix}_{PrivateKey_a}$$

Figure 2.11. It is the formulated version of the first step. (a: prover, b: witness) [47]

2. The proving node receives the answer that the witness node generates for the location proof. While the proving node proves this response; the geographic location must be within short-range communication range, only the peers who received the request must respond, and must be signed with the private key of the witness node [47].

$$Response_{b \to a} : \begin{bmatrix} Request_{a \to b} \\ PublicKey_b \\ (Lattitude, Longitude)_b \\ timestamp \end{bmatrix}_{PrivateKey_b}$$

Figure 2.12. It is the formulated version of the second step. (a: prover, b: witness) [47]

3. The proving node digitally signs the proving block by putting all the location responses, its own public key, and the hash of the last block into a block. Consensus is reached if, after this block is posted on the network, the majority adds it to the chain. If not, the block is canceled and not added to the chain [47].

$$Block_x : \begin{bmatrix} Response_{b \to a} \\ Response_{c \to a} \\ . \\ . \\ . \\ Response_{d \to a} \\ PublicKey_a \\ h(Block_{x-1}) \end{bmatrix}_{PrivateKey_a}$$

Figure 2.13. It is the formulated version of the third step. (a: prover, b: witness) [47]

Although this consensus protocol is still in its infancy in terms of blockchain networks, there are two important applications called FOAM and XYO that use this technology [46].

2.3.3 Proof of Delivery (PoD)

The actors of the classical process of sending any item from one place to another for any purpose are the sender, the organization providing logistics services that contribute to the transportation of this item, and the consignee who will receive this item. Due to the nature of this process, the fact that the actors are located in different places requires proof that this product has been delivered. Proof of delivery shows that the product has reached the rightful owner and informs all parties involved about the delivery status [48].

In the literature review, some PoD designs do not have a reward or fine mechanism to encourage stakeholders to be honest. The goodwill of the actors involved in the product delivery is relied upon. In some other design structures, a reliable third-party entity is trusted, in short, the money for the product is under the control of this independent entity until the product reaches the buyer. If there are no problems, the money is transferred to the seller's wallet. This system not only contradicts the decentralized structure of the blockchain but is also a costly method because the independent central structure provides this service for a fee [48].

In the implementation of this concept in the blockchain, concepts such as liability, fine and reward, auditability, integrity, authentication and authorization, timebound, and off-chain arbitration are very important for traded physical assets. The ability to attribute certain actions to a particular actor is considered a liability. Rewards and fines are used to encourage participating organizations to behave honestly; otherwise, they will be punished. Auditability is when the system provides a mechanism to track and trace back actions in a completely safe and reliable manner. Integrity means that all transactions are time-stamped and resistant to manipulation. Authentication and authorization enable specific actors to perform specific tasks and operations and prevent unauthorized actions. Timebound is about the item reaching its final destination within a certain period of time. Additionally, off-chain arbitration is another important requirement that allows the adjudicating body to resolve any dispute in case any party makes false claims. The selected arbitrator may have full access to the funds that may be distributed by the arbitrator after reviewing the evidence in the ledger [48].

2.4. Hyperledger

Hyperledger has been created to develop open-source, collaborative, and cross-industry blockchain technologies. In this formation led by the Linux Foundation; it is a global collaboration involving top companies in the fields of technology, supply chains, manufacturing, banking & finance, and IoT. Hyperledger hosts more than 230 organizations within its structure. It has released 10 different projects. These projects are still under active development. The main positive aspects of Hyperledger are that it paves the way for better productivity through specialization, prevents labor consumption by preventing repeated efforts, and provides better quality control of the code based on its open-source structure [49].

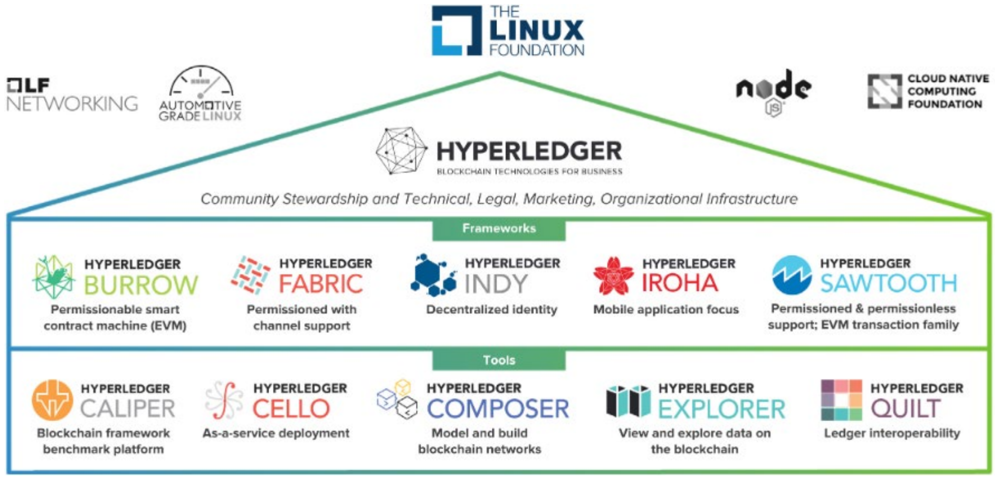


Figure 2.14. The Hyperledger structure [49]

In distributed architectures, there may be very different levels of requirements for each use case. Each company has different needs in the form of management (decentralized or centralized), block-adding time, security, and many different parameters. For example, in cases where the level of trust between users is high, a protocol that can add blocks to the network faster can be selected as a consensus protocol in order for the blockchain to work faster. Conversely, in cases where the level of trust between users is low, a consensus protocol with higher security measures can be chosen. Although this may slow down the operation of the network, it will meet the expectations of the users as it also increases its security. In order to respond to diversity, as in this example, all Hyperledger projects (Figure 2.14.) are designed to include features that are modular, very secure, interoperable, independent of cryptocurrencies, and completed with APIs [49]. A comparison of various features of Hyperledger frameworks is given in Table 2.2.

Table 2.2. - Comparison of properties of Hyperledger frameworks (adopted from [50])

Properties	Fabric	Indy	Sawtooth	Iroha	Burrow
The ability to modularize	High	Medium	High	Low	Low
The ability to be flexible	Medium	Medium	Medium	High	Medium
The ability to scale	Low	Medium	High	Low	Medium
Membership service	✓	×	×	×	×
Decentralized identity	×	✓	×	×	×

2.4.1. Hyperledger Fabric

Hyperledger Fabric is an open-source blockchain framework designed for use at the enterprise level. It is a distributed ledger that focuses on modular architecture, privacy, flexibility, and scalability. Due to these features, it is versatile and has a structure to meet the needs of every sector. Supporting multiple consensus protocols, Hyperledger Fabric can be adapted to various usage scenarios.

One of the most distinctive features of this blockchain framework is that it has no relationship with any cryptocurrency. It simply tries to benefit from the benefits of blockchain technology. Having a blockchain structure independent of cryptocurrencies also reduces the risks of some malicious activities [51].

In Fabric, users can be divided into various groups and these groups can operate in separate channels. With the channel structure, competing organizations can carry out their transactions on different distributed ledgers, so they do not have to announce all their data to each other [49].

In Fabric, smart contracts can be coded in Go, Java, or JavaScript programming languages. This is very valuable as it will not require many organizations to reinvent the wheel to develop smart contracts [51]. It is also possible to version these smart contracts.

Also Hyperledger Fabric; is supported by leading cloud service providers such as IBM, Oracle, AWS, Azure, Google, Alibaba, Baidu, Huawei, SAP, and Tencent. It is the presence of industry leaders, the support of academia, and individual contributions that enable this blockchain framework to evolve continuously and rapidly [52]. The following subsections highlight some important aspects of choosing Hyperledger Fabric for aid delivery in this study.

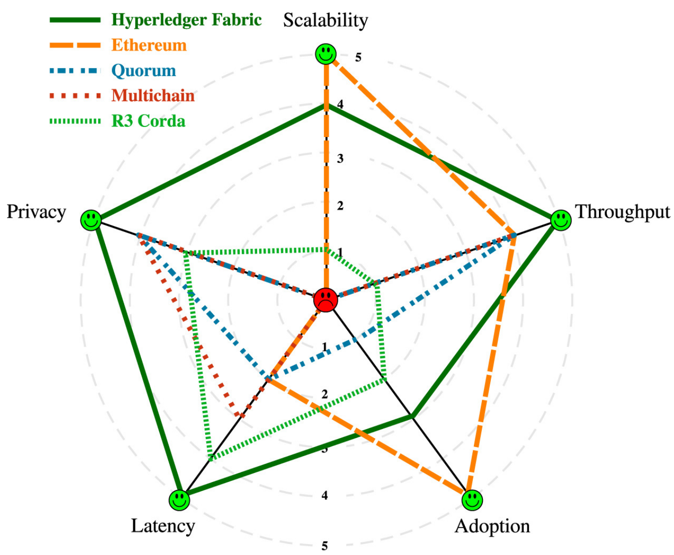


Figure 2.15. Overall analysis of some blockchain frameworks [53]

Hyperledger Fabric is chosen within the scope of this thesis because of its features such as permissioned membership, performance, scalability, adjustable privacy, security, pluggable modular architecture, and support from a strong community [54]. Figure 2.15. includes a radar chart showing the behavior scores of blockchain frameworks according to various parameters. Based on the average of each parameter in this graph, Hyperledger Fabric is the blockchain framework with the highest score. This does not simply mean that Hyperledger Fabric is the best blockchain framework, but it is an important graph that shows which parameter is a more important feature for which framework. At this point, system designers need to choose which parameters they can compromise on and proceed accordingly [53].

Permissioned Membership

In Hyperledger Fabric, it is known who all the participants in the network are. When viewed with this structure, it is suitable for various institutions, organizations, and firms, as there is no claim to anonymity [49]. In an environment where all participants are known, the risk of a malicious attack via a smart contract is significantly reduced [51].

Performance and Scalability

Hyperledger Fabric's modular architecture has a workflow layout consisting of three parts. First, there is the distributed logic processing and agreement (smart contract/chaincode) step, the second step is transaction ordering, and the last step is transaction validation and commitment. Separating the workflow in this way; Node types have the benefits of requiring less trust and verification between each other, being scalable, and providing optimized performance [49]. There is also an ongoing working group specifically for performance and scalability in Hyperledger Fabric [54].

Adjustable Privacy

Any organization may need to hide various data. The reason for this need is that these could be things like inter-organizational competition, protection laws, or the need for personal data to remain confidential. In this architecture, data flows through “channels” only to the parties that need to know the data [49].

This blockchain framework also addresses a very critical problem that other frameworks do not address. As mentioned in the previous sections, the blockchain consists of blocks added on top of each other, and in fact, it is in its nature that these blocks cannot be changed and cannot be deleted. However, this unchangeable and undeletable situation conflicts with the right to be forgotten and other data protection laws in different countries (GDPR in Europe, KVKK in Türkiye). Hyperledger Fabric has some solutions that will ensure that data is stored in accordance with these laws [54].

Pluggable Modular Architecture

Hyperledger Fabric's modular structure provides versatility. In this way, it is possible to keep up with many differences on a sectoral basis. Within this blockchain framework, there is a modular structure that allows the addition of components required in the design of the network. These necessary things may later become unnecessary for the network, and these components can be replaced to prevent problems in the system [49].

Security

Effective authentication, protection, and management of digital keys in Hyperledger Fabric are provided by HSM (Hardware Security Module). In areas related to identity management, HSM increases the level of protection of keys and sensitive data [49].

Community Support

Supported by the Linux Foundation, the project also has very high-quality documentation and many additional supporting resources [54].

2.4.2 Components of Hyperledger Fabric

The basic components of Hyperledger Fabric are one of the basic building blocks necessary for the healthy functioning of the network. These components include crucial elements such as CA (Certificate Authority), channel, ordering service, orderer, organization, and peer. Each of these parts performs different tasks to ensure the security, integrity, and accuracy of transactions on the network. Below, the basic components of Hyperledger Fabric are described [55].

CA (Certificate Authority)

CA has a high level of importance to a network because it distributes X.509 certificates that can be used to identify components belonging to an organization. Certificates issued by CAs to indicate that an organization has approved the transaction outcome are a prerequisite for a transaction to be accepted on the ledger [55].

The structure known as MSP (Membership Service Provider) ensures that certificates are matched with member organizations. This structure creates an MSP that is linked to a root

CA certificate, verifying that its components and identities were created by the root CA [55]. An organization's MSP determines which CAs that organization can and therefore cannot trust. When an organization's MSP is added to the channel, this means that the organization is approved by network members. The difference between CA and MSP is that CA creates certificates that serve as a kind of identity, while MSP contains a list of these identities [56].

Peer

One of the main components of the Hyperledger Fabric blockchain network is peers. Peers are very important because they host ledgers and chaincodes (smart contracts). Peers also manage transaction proposals and responses to them. Peers can be easily created, started, stopped, reconfigured, and deleted. Consistent data copies and chaincodes between peers in a channel provide redundancy in the network, eliminating single-point failures and keeping ledgers constantly up to date [57].

Channel

Peer nodes, ordering nodes, and applications are the components that make up the channel. They agree to collectively manage and share copies of the ledger by joining a channel. Channels are a tool that allows a specific application group, peers, and organizations to communicate with each other on a network [57].

A peer has the capacity to host multiple ledgers, which refers to a flexible system design in which a single peer can belong to multiple channels within a Fabric network. In the most primitive form, a peer hosts only a single ledger and is therefore connected to only a single channel. However, there are also complex situations such as hosting more than one ledger if a peer can belong to more than one channel [57].

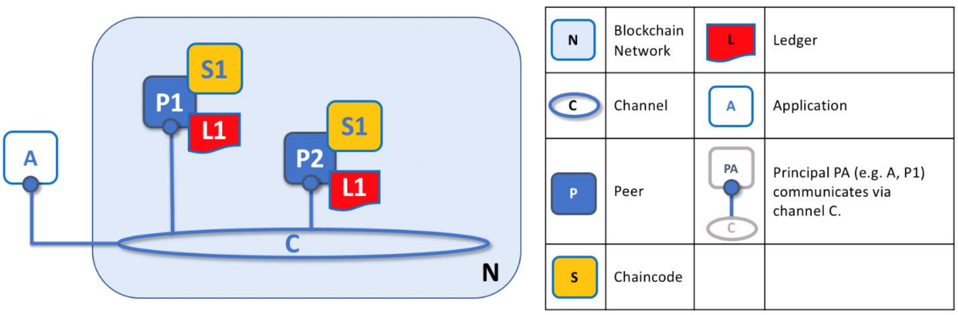


Figure 2.16. To communicate with peers P1 and P2, application A uses the gateway service on channel C [57]

Organization

In Hyperledger Fabric, networks are managed by a group of organizations rather than a single organization. An ideal Hyperledger Fabric blockchain network is built and managed by multiple organizations that provide resources to it. These resources include peers, chaincodes, and ordering service nodes. What is meant by the ideal chain is that the network does not fully exist without organizations incorporating their own resources into the collective network. That is, when collaborating organizations include their own resources in the network; the network grows, and the durability and security of the network increases. Organizations can decide how to use the data in the ledger, so the applications may not be the same. Therefore, even though each peer maintains an equal copy of the ledger, the activities of the organizations on the network may differ [57].

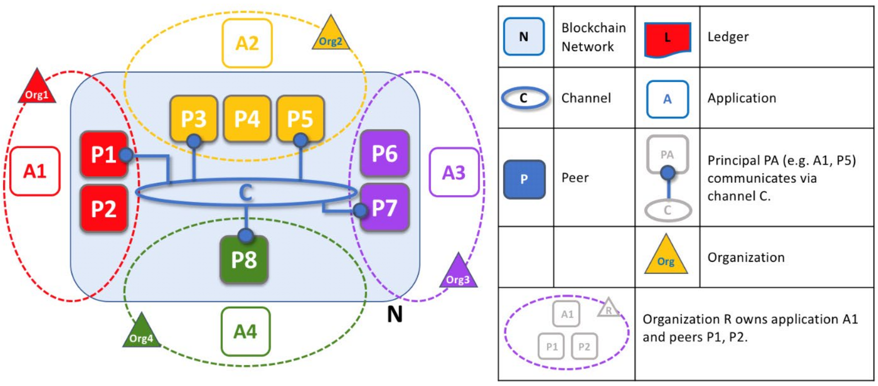


Figure 2.17. A diagram showing the inclusion of the organization structure [57]

Ordering Service

Orderers are what keep the ledger consistent and up-to-date across the channel. The ledger update process is not like querying, because the ledger cannot be updated with a single peer; other peers must also be involved in the consensus mechanism. Peers and orderers carry out consensus processes together [57]. Orderers who provide in-channel authorization determine who can read and write data and how these authorizations are configured. In this case, the orderer processes the configuration update to ensure that the requester has the appropriate administrative authority.

The workflow process of orderers, that is, the operation of the consensus mechanism, consists of three main steps, which are explained respectively below:

1. Transaction Proposal and Confirmation

First, a transaction proposal is sent to the Fabric Gateway service via a peer. This peer executes the proposed transaction or forwards it to another peer in its organization to execute this transaction. Fabric Gateway also forwards it to peers in relevant organizations in accordance with its approval policy. The peers that perform the confirmation process execute the transaction and send the incoming transaction response back to Fabric Gateway [58].

2. Transaction Submission and Ordering

After the first step is completed, the client application receives a confirmed transaction proposal response. Fabric Gateway transmits this confirmed transaction along with other confirmed transactions to the ordering service, which orders them and packages them all into one. Nodes that are special for the ordering service execute a large number of transactions at the same time. Multiple channels share the ordering service consisting of these nodes [58].

The blocks saved to the ordering of one's ledger are distributed to the peers in the channel. Inactive members can communicate with active members to transfer block information.

Each confirmed transaction is packaged as a block when it reaches the ordering service. Ordering services, except channel configuration operations, do not interfere with the content of transactions. Committed or verified transactions cannot be undone or canceled [58].

3. Transaction Verification and Commitment

The last step is to send the ordered and packaged from the ordering service blocks to the channel peers for verification and commitment to the ledger.

First, ordering service blocks are distributed to all channel peers. Each peer independently verifies distributed blocks to ensure consistency across ledgers. After this verification session, transactions found to be invalid are still held in the unchanging block created by the orderer, but the peers mark them as invalid and do not update the state of the ledger [58].

2.5. Use of Blockchain in Humanitarian Assistance

While carrying out humanitarian assistance activities, the use of blockchain-based technology instead of traditional database management systems is an approach that can make a difference in terms of data security and anonymity, as the data of beneficiaries may contain sensitive data such as personal information.

One of the important projects that bring together blockchain and humanitarian aid is the Ethereum-based project called "Building Block", which was implemented by WFP (World Food Programme) initially in Jordan and Bangladesh, and later in Lebanon and Ukraine [59]. Thanks to the project in which more than 4 million people are beneficiaries per month in the Jordan and Bangladesh phase of the project, 3,5 million dollars of bank transaction fees have been saved. In Ukraine phase, 18 humanitarian organizations participated in Building Blocks between May 1 and August 31, 2022. Thanks to this cooperation, exactly US$35 million (UAH 1.2 billion) of unwanted aid overlap has been eliminated [60]. This output is very important to understand how great resource conservation this type of cooperation brings.

Building Block is blockchain nodes with computer servers that are operated independently by each participating organization. These organizations have come together on an independent humanitarian assistance blockchain in order to realize transactions in real time, connect with each other, and send data securely to others. All the organizations involved in this blockchain that have been created are equal owners of this blockchain, and all these organizations play an equal role in the maintenance of the blockchain. With this network, institutions or organizations can keep a record of the services provided to beneficiaries, which prevents the waste of aid resources, the labor consumption of volunteers and professionals in this field, and most importantly, preventing those in need from not receiving assistance [61].

When a crisis breaks out, the institutions or organizations intervening in the situation will likely provide assistance to the same people in need, usually each institution or organization has its own way of working. It has been tried to increase the transparency and effectiveness of assistance activities in the Building Block network. In this blockchain, it is possible to track and coordinate many different types of assistance, including cash, food, WASH (water, sanitation, and hygiene), medicine, and other things. Thanks to this network, households can receive assistance from different humanitarian organizations at once for different types of basic needs (Figure 2.18). In this way, access to humanitarian assistance becomes easier. In addition, sensitive data such as personal data and biometrics are not stored in this network, and the security and privacy of the beneficiaries are ensured with anonymous identifiers [61].

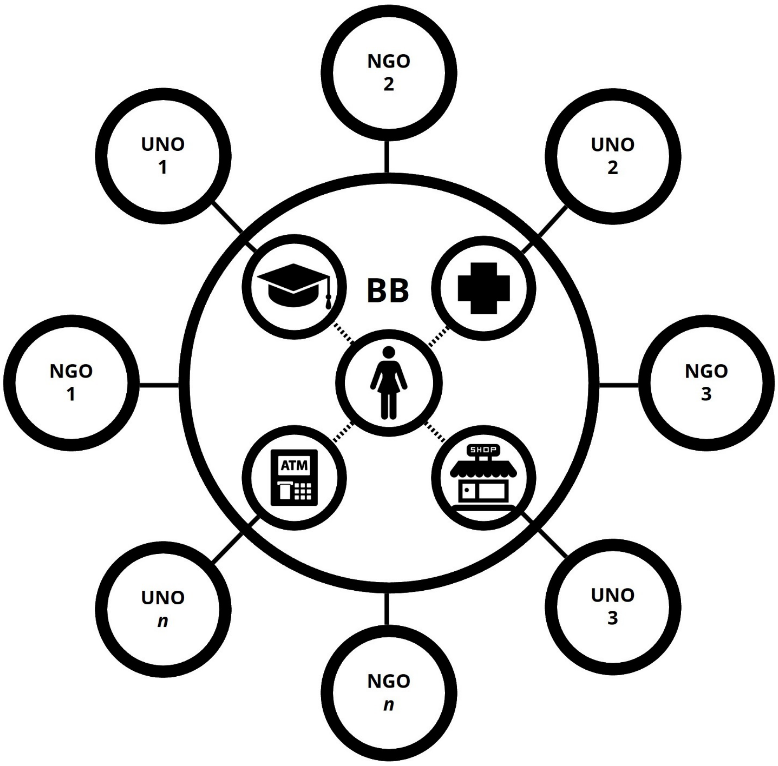


Figure 2.18. BB: Building Blocks, UNO: United Nations Organization, NGO: Non-Governmental Organization [62]

Blockchain technology is shown as one of the most effective technological infrastructures that helps to respond quickly to people who are in a situation of disaster-related victimization and reduce time losses [63]. In disaster situations, lack of coordination does not bring any benefit, but sometimes it also causes harm. This lack of coordination is one of the reasons for the distrust of donors. The blockchain, which provides great benefits in stock tracking and monitoring the flow of aid, is very effective in responding to immediate needs in emergency situations [64]. There are also studies indicating that the use of smart contracts increases the effectiveness of aid activities in increasing trust and transparency between humanitarian aid organizations and the companies that provide services in the field of logistics, which are business partners of these organizations [65].

Proposals for a disaster recovery-based blockchain design are available in the literature. While making this design, the authors identified the problems that the aid sent to the disaster victims did not actually reach the targeted places, the aid was not distributed fairly, and in some cases, it was not possible to deliver the aid to its target, that is, the disaster victim (due to situations such as roads being closed to traffic). The aim of the study is the design of a system for the distribution of aid to disaster victims. In this system, it is about monitoring the aid and verifying the stages that the aid passes through on its way to its target [66]. In this way, it is desired to prevent waste and aid items from being seized by ill-intentioned people [67]. It was thought that thanks to these precautions, people would be more willing to help and provide aid. In addition, another goal is to deliver aid to disaster victims even when the roads to be used for aid delivery are closed. The authors tried to ensure that the aid is sent to the right person with trust and justice via blockchain technology. Hyperledger Fabric was used as the blockchain framework in the application area. In the tests carried out for the study, it was seen that the performance of the blockchain is efficient if the appropriate configurations are set, even if there is a large aid distribution activity [66].

Within the scope of the study, the actors who will take part in the blockchain network are donors (individuals), governments, aid NGOs, service providers, insurance companies, auditors, field workers, devices (UAV and other IoT devices), and disaster victims [67].

Table 2.3. Roles of actors (adopted from [67])

Party Type	**Initiator**	**Node**	**Node**	**Node**	**Node**	**Node**	**Receiver**
Actor	Donor	NGO	Logistic compan y	Air transport company	Disaster coordinati on center	UAV compan y	Victim of disaster
Worker		NGO warehouse employee	Carrier	Airport staff	Crisis managers	UAV operator	
Sensor		NGO warehouse exit sensors	Barcode scanner truck GPS	Plane loading docks	Disaster coordinati on center unloading docks	UAV cameras , GPS	

The main duties of the actors in this study are briefly mentioned in Table 2.4 through a scenario.

Table 2.4. Task of actors (adopted from [67])

Actor	**Main Task**
Donor	The goal of a donor's donation to the NGO is to aid those affected by disasters.
NGO	An NGO purchases supplies or gets its current in-kind aid ready for truck delivery to the airport.
Logistic company	The stuff from the charity organization is transported to the airport by a logistics company.

Actor	Main Task
Air transport company	The in-kind aids are brought to the disaster coordination center by an air transport business, which picks them up from the trucking company and drives them to the airport.
Disaster coordination center	The in-kind aids were delivered by air to a disaster coordination center, which then readied them for distribution.
UAV company	The disaster coordination center provides in-kind aid to the UAV company, which then distributes it to the disaster victims.

At various stages of the workflow in this proposed blockchain, the in-kind item is tracked with tools such as sensors, barcode readers, GPS devices, and cameras. Table 4 shows the types of data obtained with these tools [67].

Table 2.5. Sensor outputs (adopted from [67])

Sensor	**Data obtained**
NGO warehouse exit sensors	In-kind aid tracking information Truck entry and exit information
Barcode Scanner, Truck GPS	Truckload information (in-kind aid tracking) Truck GPS coordinates
Plane loading docks	Plane load information (in-kind aid tracking)
Disaster coordination center unloading docks	Disaster coordination center in-kind aid receipt/tracking
UAV GPS	UAV location coordinates
UAV Cameras	UAV activity images (images before and after the drop)

Another application of blockchain technology is a framework in which people in need and aid providers communicate in case of disaster. All users in this system can monitor the processes of providing aid. In addition, there is also an incentive system for aid providers in the system, as in a classic blockchain structure, for their services and honest behavior. The following scenario can be given as an example of the operation of the system: A person in the disaster area is seriously injured in the ankle and reports this serious injury via the mobile application. The system sends this request for help to the relevant members, i.e. doctors. Doctors who want to accept this request report this to the network. This network selects the best among the doctors who agree to help, based on their distance to the person needing help and their service history. With a framework of this nature, time delays in rescue and aid activities are tried to be minimized, possible corruption situations during these activities are prevented and a transparent network structure is encouraged [63].

2.6. Geospatial Data in Blockchain

The use of geospatial information in blockchain databases is increasing where the first examples are from land registry and cadastre. Transparency in land registry is essential since there is mistrust between parties in land registration transactions. According to research, the technology component takes the position of the weakest party, the land registration officer. Put another way, fraudulent transactions can be avoided by substituting a consensus copy of the blockchain for the officer. According to the report, land registration systems based on blockchain technology have been initiated in Estonia, Honduras, Georgia, Ghana, and Sweden; among these, Sweden has advanced significantly with a pilot project [68]. Distributed peer-to-peer (P2P) systems also have the ability to eliminate intermediaries and rely on third parties for navigation and other geographic components. Landowners who register their properties on a blockchain may be able to have a unique representation; in other words, they will have access to a private encryption key [69]. There are examples of blockchain technology which can be used to create a platform for crowdsourced geographic data sharing [70]. In addition, important organizations such as the Open Geospatial Consortium, NASA, and the European Space Agency have started to use blockchain technology in levering and distributing spatial data.

Additionally, according to a study conducted by Mendi et al. [71] for Türkiye, a blockchain-based land registry system has been created. In this system, issues such as sharing of map data, cadastral data management, property rights, protection of property rights in geological resources, management of water resources, and air pollution measurement are addressed [72]. There is also an approach that proposes a blockchain-based system for decentralized sharing of remote sensing data. This system not only provides reliable data but also guarantees data integrity [73].

2.7. Geospatial Data in Humanitarian Assistance

The use of spatial data related to humanitarian assistance provides institutions and organizations in this field with a clear perspective on issues such as decision support mechanisms, monitoring, evaluation, and organization of humanitarian assistance. Geographic information systems (GIS) are actively used by UN agencies, government institutions, and NGOs in the field of humanitarian assistance. One of the most effective examples of this is the DTM (Displacement Tracking Matrix) system of IOM (International Organization for Migration). DTM is a system designed to monitor migration movements. It is designed to regularly and systematically collect, process, and disseminate data to better understand the movements and evolving needs of migrants, on-site or en route (transit country or destination country). DTM helps decision-makers and humanitarian assistance partners maximize their information resources by turning masses of data into information. IOM provides DTM data as a common resource to actors in the humanitarian field, relevant government agencies, and other actors interested in supporting the migrant population. Within the DTM, data collection studies are developed and implemented in order to provide data flow in post-crisis interventions, as well as to create a data source for response teams at the acute stage of crises. DTM, which creates datasets that become suitable for analysis over time with regularly collected data, has also assumed a significant role in providing information to policy and research. DTM activities are managed by teams in the countries where the data is collected. The sizes of these teams show differences according to the projects. The teams are composed of professionals of different abilities, including geographical information systems specialists, information management specialists, analysts, field coordinators, and enumerators.

DTM produces many different information products within itself such as raw dataset - cleaned and verified dataset, site profile, dashboard, thematic maps / GIS products, reports [74] (Figure 2.19.), web (geo) portal, solutions, and mobility index (SMI). These information products can be explained as follows:

The data forming the datasets from these products are collected by DTM teams. Data collection can be conducted in the field or digitally. This varies depending on the dynamics of the study. If the collected data has not undergone various data cleaning processes (automatic and manual data verification, control of data quality, etc.), it is called raw data; and if cleaning and verification of the data has been performed, this is a cleaned and verified dataset. Many different actors need to make analyses of the data to plan their own activities, and in order to make a healthy analysis of the data, cleaning and verification processes must be carried out. Datasets make analysis possible and easier.

Site profiles are extremely helpful in providing an overview of the camp areas where displaced people live. Humanitarian assistance workers in the field use these documents to quickly understand the situation in the area and highlight important information that requires attention. Dashboards are very useful and smart if there is quickly changing statistical data. Having teams produce non-dynamic reports in situations where there is constantly changing data causes a waste of time and effort. When dashboards are well designed, they are an effective tool for tracking real-time or short-term data and catching trends.

DTM uses data to produce a variety of thematic maps that display information of geographical importance for planning and operational use. These are involved in a wide range of applications, ranging from simple maps consisting of only a few parameters to complex analyses combined with other data sources of DTM [74]. Thematic maps are one of the simplest and most effective tools of data analysis. Thematic maps provide an easier understanding of spatial information, statistical analysis, and maps. For example, the provincial map can be colored according to the migration density in each province. Accordingly, dark colors indicate high migration rates, while light colors indicate low migration rates. This mapping method can show the same data in various ways [75].

DTM reports analyze the collected data and provide a comprehensive summary of a specific migration event. These reports provide information for decision-making and strategic and operational planning [74].

SMI is a tool developed by the DTM unit of the IOM to measure the severity of conditions at locations. SMI helps identify the root causes of vulnerability using data to address displacement. This index aims to find lasting solutions for vulnerable communities and provides information to guide decision-makers about the dynamics of stability, peace, development, recovery, and reintegration. SMI concentrates data on four main pillars: demographic and mobility trends, safety and security, access to services and infrastructure, and social cohesion. SMI generates large data regionally, helping to identify smaller areas of stability and fragility [76]. Spatial and non-spatial data are compiled into annual reports to support decision making [74].

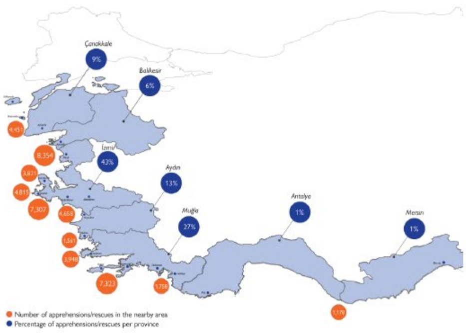
Migrants in Irregular Situation

Intercepted and Rescued Persons on Sea

According to Turkish Coast Guard (TCG) daily reports, TCG recorded **49,525*** migrants in irregular Situation at sea and registered **41*** fatalities in 2022. During the reporting period, there was an increase of 25,849 intercepted persons on sea by TCG compared to previous year. The top ten nationalities of intercepted/rescued persons are Afghan, Liberian, Syrian, Palestinian, Congolese, Bangladeshi, Yemeni, Eritrean, Central African and Iraqi.**

*Data source TCG, end of 2022

**The information on nationality provided in this report is based on the nationality declared by migrants as reported by the Turkish authorities.



Source: TCG, 31.12.2022

Note: This map is for illustration purposes only. The boundaries and names shown and the designations used on this map do not imply official endorsement or acceptance by IOM.

"Interceptions/Rescues by Turkish Coast Guard Statistic for 2022 (1 January - 31 December)"

Time Period	Numer of Cases	Number of Migrants in Irregular Situation	Number of Deaths	Number of Organizers
	All Seas	All Seas	All Seas	All Seas
January	77	1,929	1	6
February	83	2,637	2	49
March	76	2,771	0	10
April	117	3,243	2	17
May	139	4,680	0	45
June	160	4,660	1	24
July	161	5,103	1	23
August	194	6,331	3	43
September	165	5,710	22	38
October	169	4,777	6	23
November	143	3,666	1	4
December	133	4,018	2	10
Total	**1,617**	**49,525**	**41**	**292**

Annual Breakdown of Interceptions/Rescues by Turkish Coast Guard in Türkiye

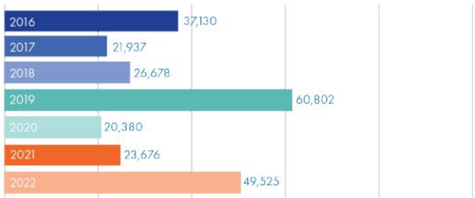


Figure 2.19. A page from the Türkiye — Migrant Presence Monitoring - Annual Migrant Presence Monitoring Report (Jan-Dec 2022) [77]

3. BLOCKCHAIN BASED AID DELIVERY

A general conceptual model for location-based smart contracts and blockchain based in-kind aid delivery is developed in this chapter. The chapter starts with the organizations and network of transactions for in-kind delivery. The items that flow in the network and associated data requirements are listed. The next section provides the roles and responsibilities within the network of organizations to support the process of in-kind aid delivery. The use cases of the actors and their activity diagrams are provided following with the discussion on location-based smart contracts to support transactions requiring location information.

3.1. Organizations and Network

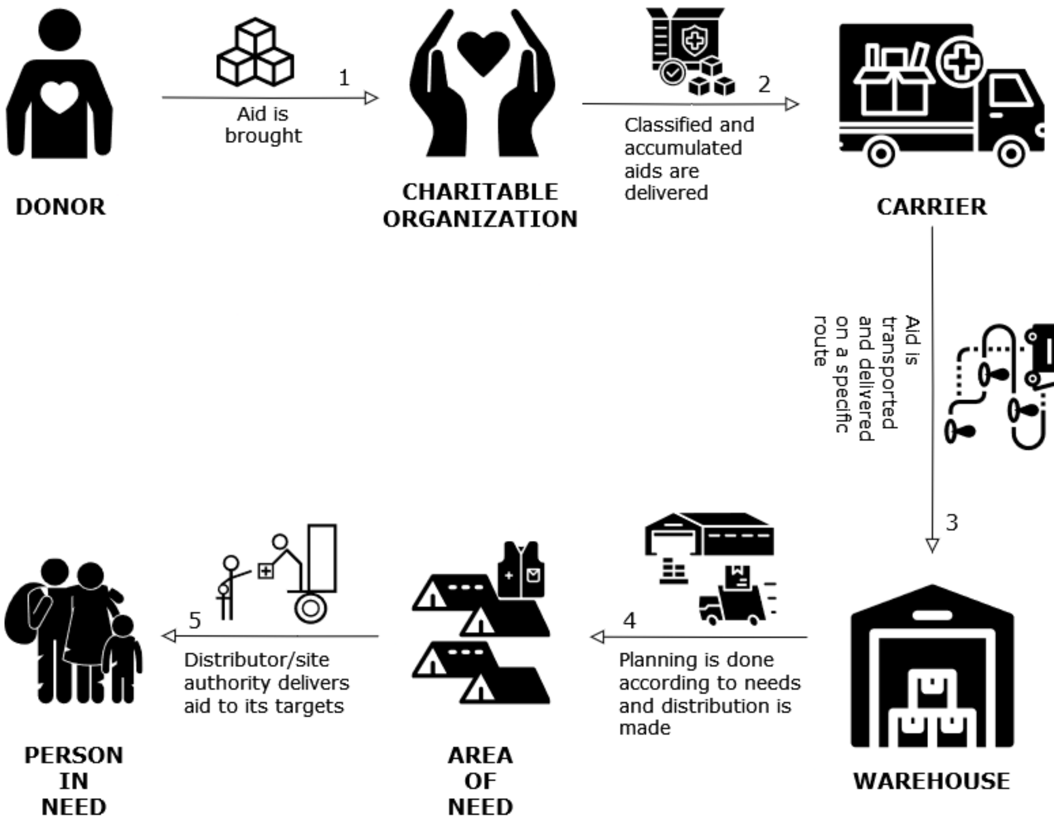


Figure 3.1. Organizations and network of transactions

The donor, who is the first link in the chain, brings the items he/she wants to the branch of the preferred charitable organization. In this step, the delivery acceptor, who is the staff of the charitable organization, learns the information about the donor's preferred aid target location/region from the donor himself and enters the information into the system. The delivery acceptor creates a unique identifier (ID) for this incoming donation and gives it to the donor for further tracking. Thanks to this ID, the donor can be able to track the location of his/her donation and view various information about the donation. Once a certain amount of aid items has accumulated within the aid plan, the delivery acceptor delivers all to the carrier to take them to the warehouse. In the context of Türkiye, the carrier may be a private company providing logistics services or aid organizations may have their own logistics fleets. Carrier delivers aid items to the warehouses by following certain routes. In the context of post-disaster emergency aid operations in Türkiye, these warehouses may belong to a charitable organization or AFAD (Disaster and Emergency Management Presidency). In this context, in Türkiye, either the warehouse worker of the disaster coordination center or the warehouse worker of the charitable organization receives the aid items delivered by the carrier. Likewise, the distributor who will distribute aid items to areas where people in need are located (container city, tent city, irregular settlement not affiliated with a formal organization, home, etc.) can either be an employee of the disaster coordination center or an employee of the charitable organization. Mass and temporary formal settlement areas are the areas under the control of the disaster coordination center. There are site authorities in these areas. If the distributor takes aid items to such an area, the person who will receive the aid item is the site authority. Site authority distributes aid items to those in need, who are the last link in the chain, ordered according to the needs in the area. If the aid is delivered by the distributor to an irregular settlement or a home, the aid items are delivered directly to the person in need. Table 3.1 shows the data that organizations share with each other throughout this workflow.

Table 3.1. Data shared during workflow

Data Item	**Properties**
In-kind aid	ID, name of the item, type of the item, brand, and model of the item (if applicable), amount of the item, price of the item (if applicable), condition of the item, expiration date (if applicable), preferred delivery location and photos of the donated item.
Location	ID, location type (delivery stop, organization warehouse, final destination, etc.), name of the area, polygon and other geographical attributes of the area, activity status of the area, and settlement type in the area.
User	ID, information about the role of the user, the name of the organization to which the user is affiliated, the department and title of the user in the organization, the name and contact addresses of the user, and the geographical attributes of the place where the user works.
Organization building	ID, type and name of the organization, type and name of the building, activity status of the building, date of commissioning of the building, staff

Data Item	Properties
	responsible for the building and their contact addresses, and geographical attributes of the building.
Transfer	ID, information about the aid to be transferred, information about the previous and next actor who owns the transferred aid item, information about the previous and next location of the transferred aid item, and transfer date.
Working Area	ID, geographical areas where users work

3.2. Roles and Responsibilities

Individual roles and their responsibilities within the in-kind aid delivery process are described below. For each role, the requirements from the system are also listed informally.

3.2.1. Donor

The donor represents individuals, organizations, or governments supporting humanitarian aid projects in a broad sense. These actors contribute to aid activities by providing financial resources, materials, or services. Donors can offer assistance to national or international humanitarian organizations, non-governmental organizations, or directly to those in need. Contributions from donors can range from post-disaster recovery efforts to long-term development projects.

In the scope of this study, the term "donor" refers to individuals supporting humanitarian projects, particularly those providing material donations. The donor can bring the materials requested to be donated to a branch of the organization far from the aid target, or directly to the organization's warehouse in the region of the aid target. The choice depends on the donor's preference and the organization's capacity. If the donor delivers directly to the warehouse worker instead of the delivery acceptor, they bypass the delivery acceptor and carrier steps, allowing the warehouse worker to assume the role of the delivery acceptor. After delivering the aid, the donor can monitor its spatial location anytime using the client provided by the organization, ensuring the aid reaches the correct destination in time.

3.2.2. Delivery Acceptor

The personnel working at a specific branch of aid organizations typically accept the donated materials provided by the donor in the next stage. These individuals record, inspect, classify, and store the incoming materials. They also assess the type of needs addressed by the materials and create distribution plans. The location information is generally checked and inserted by this actor where a location-based smart contract may be required.

3.2.3. Carrier

Carriers are responsible for transporting aid materials from the organization's branch that accepts delivery to the target area. They manage transportation vehicles, determine routes, ensure the safe and efficient transportation of materials, and handle possible disruptions in the transportation process. The carriers follow predetermined main and alternative routes, performing location-based transaction control at key points along the route.

The carrier can take the aid materials received from the organization's branch to three alternatives. The first is the organization's warehouse located near the aid target. The next two alternatives may be pursued for two reasons. One reason is the absence of an organization warehouse near the target area, and the other is the urgency of the situation, in which case the aid is delivered directly to either the formal site authority or the aid target. In the case of sending the aid to a formal site managed by government authorities, the aid is received by the site authority. If the aid is sent to an informal site not managed by government authorities or a disadvantaged group that has migrated to its current location for any reason and does not live in any migration center, then the aid is directly received by the person in need. This kind of transaction requires a smart contract capable of checking the intent of the donor for in-kind aid delivery.

3.2.4. Warehouse Worker

Before reaching the target area, aid materials are stored in the organization's warehouses after being received from the carrier. Warehouse workers are responsible for storing materials, managing inventory, performing packaging processes, and making the materials quickly accessible in case of emergencies. Warehouse workers prevent the loss of aid supplies and speed up the distribution process by creating effective inventory management and warehouse layout. This phase is critical to managing supplies and preparing for distribution effectively.

If the donor directly delivers to the warehouse worker, the warehouse worker acts similarly to the delivery acceptor (excluding the handing-off task to the carrier).

3.2.5. Distributor

After aid materials reach the target area's warehouse, the distribution phase begins. Distributors are the organization’s local representatives in the region responsible for delivering aid materials to those in need. These actors collaborate with local communities to identify needs, create distribution strategies, and ensure a fair aid process by effectively distributing materials. Distributors play a crucial role in ensuring that aid reaches the most needed points by taking into account social and cultural factors.

The distributors can take aid materials received from the organization's warehouse to two alternatives. In one of these two alternative situations, if the aid is sent to a site managed by government authorities (formal site), the aid is received by the site authority. If the aid is sent to an informal site (not managed by government authorities) or a disadvantaged group that has migrated to its current location for any reason and does not live in any migration center, the aid is received directly by the aid target themselves. The distributor is also responsible for location-based control of in-kind items to reach their intended location.

3.2.6. Site Authority

Authorized individuals or organizations managing the region where aid materials are distributed come into play at this stage. For example, in a disaster area like a container city, these authorities are responsible for ensuring orderly and effective aid distribution, managing security, and securing the general welfare of the community. They also identify and report deficiencies in formal sites to relevant authorities for correction/improvement. After identifying deficiencies, site authorities are obligated to distribute incoming aid fairly among those in need.

3.2.7. Person in Need

In the last step, the focus is on the primary purpose of aid. The person in need represents individuals or communities to whom aid is intended to be delivered. This actor includes individuals in disaster areas in urgent need as well as communities that have migrated for any reason and find themselves in disadvantaged situations. Aid is directed towards meeting basic needs, improving living conditions, and supporting post-disaster recovery for these targets.

The person in need can receive aid in three different roles depending on their location and urgency. If the person in need resides in a formal site, they receive aid from the site authority. If they stay in an informal site, they can receive aid from the carrier or distributor. If the person in need of assistance is a disadvantaged group who has migrated to his current location for any reason and does not live in any migration center, he/she still receives aid items from the distributor or carrier.

3.3. In-kind Aid Delivery Process

The activities of each role within the proposed in-kind aid delivery framework are given in Figure 3.2. The location tracking gates are logic that shall be checked by smart contracts to support human errors in the delivery process. The activities of each role are given below in respective subsections.

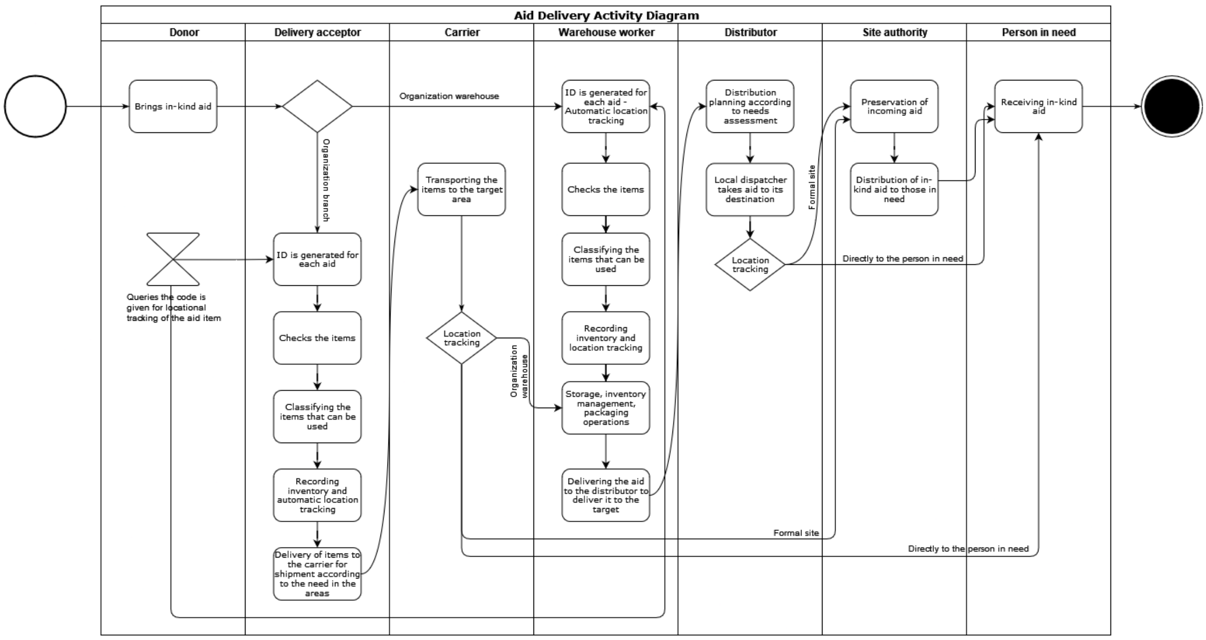


Figure 3.2. Aid delivery activity diagram

3.3.1. Activities of Donor

Brings in-kind aid

The entire process starts with a step where people who want to help other people in difficult situations but cannot deliver in-kind aid to those in need or who do not want to provide their in-kind aid with their means, bring the in-kind aid they want to donate to the branch or warehouse of any relief organization.

If the person is going to start the process of providing in-kind aid to those in need from a distant location, the donor can bring the aid to the branch of the organization that is located far from the person in need. If the person is in an area close to those in need, they may also choose to bring the aid directly to the organization's warehouse.

3.3.2. Activities of Delivery Acceptor

ID is generated for each aid

An ID code is generated for the donation. This code is required for the donor to make inquiries later. Thanks to this generated code, the donor can view the inventory information of the items donated at any time and track their geographical location. This ID creation step also provides a means of location tracking. Since the buildings of institutions or organizations will be included in the system, the location tracking process is automatically started with this step.

Checks the items

When the delivery acceptor receives the aid, it checks the condition of the items after generating the ID code. If there is an initial problem with the donated items, the delivery acceptor does not accept the items and returns them to the donor. There may also be problems with the items that cannot be detected with a superficial check. If these are noticed at later stages, the donor is informed about this, and these items are not sent to people in need.

Classifying the items that can be used

After a superficial check of the in-kind aid delivered by the donor is made, these items begin to be classified in order to be recorded in the inventory. Sending items from donors in a mixed manner may cause irrelevant items to be sent to people in need or may cause the things they need not to be delivered.

Recording inventory and location tracking

At this stage, all necessary information about the in-kind aid is recorded in the "In-kind aid" structure. Some information here is mandatory, some is optional. For an effective aid network, it is important to know how much of each item, and inventory records provide this. Additionally, within the scope of this task, location tracking is done for the first time.

Delivery of items to the carrier for shipment according to the need in the areas

The in-kind aid prepared in the organization branch according to the needs of the regions where the needy are located is given to the carrier so that they can be delivered to the target region.

3.3.3. Activities of Carrier

Transporting the items to the target area

The carrier can have three alternatives as target areas. This varies depending on the urgency, the capacity of the organization, or the situation. One of the carrier's options is to take the aid to the organization's warehouse. Another option is for the carrier to deliver the in-kind aid to the formal site. The reason for this option may be related to the inadequacy of the organization's capacity, or it may be related to urgency or the operational preference of the organization. The last option is to deliver aid directly to those in need. The reason for this option may be due to the insufficient capacity of the organization, or it may be related to urgency or the operational preference of the organization. In addition, if aid is provided to an informal site or a region where there is a direct migrant community, it is not technically possible to deliver it to the formal site authority, so the aid is delivered directly to those in need. Whichever of these three situations the carrier does, it is mandatory to perform location tracking.

3.3.4. Activities of Warehouse Worker

ID is generated for each aid

If the donor brings the donation directly to a warehouse of the organization near the target location, rather than to a branch of the organization far from the target location, the warehouse worker essentially acts as a delivery acceptor. By doing so, the donor bypasses two roles. In other words, An ID code is generated for the donation, just like the delivery acceptor does. This code is required for the donor to make inquiries later. Thanks to this generated code, the donor can view the inventory information of the donated items at any time and track their location.

Checks the items

In this step, when the warehouse worker receives the aid, just like the delivery acceptor does, the warehouse worker checks the condition of the items after generating the ID code. If there is an initial problem with the donated items, the warehouse worker does not accept the items and returns them to the donor. There may also be problems with the items that cannot be detected with a superficial check. If these are noticed at later stages, the donor is informed about this, and these items are not sent to people in need.

Classifying the items that can be used

After a superficial check of the in-kind aid delivered by the donor, just like the delivery acceptor does, these items begin to be classified in order to be recorded in the inventory. Sending items from donors in a mixed manner may cause irrelevant items to be sent to people in need or may cause the things they need not to be delivered.

Recording inventory and location tracking

Just like the delivery acceptor does, at this stage, all the information that may be required about the in-kind aid is recorded in the "In-kind aid" structure. Some information here is mandatory, some is optional. For an effective aid network, it is important to know how much of each item, and inventory records provide this. Additionally, within the scope of this task, location tracking is done for the first time (if the donor brings the in-kind aid to the organization's warehouse first).

Storage, inventory management, packaging operations

From this stage on, the warehouse worker does different things than the delivery acceptor. This duty includes ensuring that in-kind aid is stored under appropriate conditions, storing in-kind aid in an orderly manner, and carrying out final packaging before delivering it to the person in need.

Delivering the aid to the distributor to deliver it to the target

In-kind aid is delivered to the distributor.

3.3.5. Activities of Distributor

Distribution planning according to needs assessment

Distributors identify needs by collaborating with local communities. While needs can sometimes be determined based on the results of a comprehensive site assessment, sometimes it can be shaped according to simple interviews with government institutions, local and international NGOs, and local communities. An ideal needs assessment process is a blend of these two. The results of these needs assessment studies are shared with distributors, and distributors create plans to distribute in-kind aid. These plans developed based on needs assessment, play an important role in equitably distributing aid.

Local dispatcher takes aid to its destination

The organization's local dispatcher can deliver relief items to two different options. If the distributor delivers aid to a formal site, the distributor delivers the aid to the site authorities. If the distributor takes the aid to an informal site or to a place where migrants live, this time the distributor delivers the aid directly to those in need. The distributor is also responsible for location tracking based on where the in-kind aid is delivered.

3.3.6. Activities of Site Authority

Preservation of incoming aid

Some of the in-kind aid received by the distributor can be distributed immediately, but some may consist of items unsuitable for immediate distribution. For this reason, site authorities must protect items that will not be distributed immediately. Various security measures must be taken to prevent the in-kind aid planned to be given periodically from being seized by malicious people.

Distribution of in-kind aid to those in need

Site authorities are responsible for distributing aid fairly on formal sites. Within the scope of this duty, site authorities deliver aid delivered to them fairly to those in need.

3.3.7. Activities of Person in Need

Receiving in-kind aid

The people in need receive aid. As mentioned before, these people can receive assistance from three different roles. These are carrier, distributor, and site authority. In which situations and from which role it is delivered has also been mentioned in the previous sections.

3.4. Use Cases of Location-Based Smart Contracts

Due to its suitable properties mentioned in the previous sections, Hyperledger Fabric can be used as the technical infrastructure in the context of this thesis study. Three potential areas of assistance are identified. These are “Aids Provided to Formal Sites in the Disaster Area”, “Aids Provided to Informal Sites in the Disaster Area” and “Aids Provided to Migrants in Vulnerable Situations”. The first two of these defined assistance areas cover the need for emergency aid in the early post-disaster period, then improving conditions and ultimately restoring conditions. The last assistance area covers migrant individuals who have been displaced for reasons they do not want and are trying to live in disadvantaged conditions. Aid delivery operations serving different purposes may have different actors, although a majority of the activities are similar. Having different channels for different humanitarian operations will provide a healthier data structure and isolation. Moreover, this does not prevent an organization from joining in more than one channel. Thus, the same organization can work flexibly by using separate channels for different business processes or datasets.

In Hyperledger Fabric, different authorizations can consist of different actors within the same data structure. This authorization and management of blockchain is carried out using a smart contract (chaincode in the context of Hyperledger Fabric). This structure ensures that certain actors can edit or only view certain fields. The actors introduced in the previous chapters, delivery acceptor, warehouse worker, and distributor are employees of aid-related organizations, and they can be considered employees of the same organization in different departments. Although these may vary depending on the situation in each country, the site authority may also be included. In Türkiye, this is not the case. Among the mentioned potential assistance areas, the authority in formal sites is AFAD or other governmental institutions, while in migrant camps it is the Presidency of Migration Management. The proposed authorization structure is designed according to the aid distribution processes in Türkiye. What actors can and cannot do is discussed in the "Abilities of Actors and Restrictions" section below.

3.4.1. Formal Sites and Informal Sites in the Disaster Area

Two channels will be examined under this title. The "Aids Provided to Formal Sites in the Disaster Area" channel is for aid sent to formal sites established in the disaster area. The concept of formal sites means sheltering areas in the disaster area where the management is located at AFAD in the context of Türkiye. The other channel, "Aids Provided to Informal Sites in the Disaster Area", is for aid sent to informal sites established in the disaster area. Informal settlements are self-settled locations of varying composition and size, including clusters of tents, makeshift shelters, use of public buildings, or other forms of community shelter. Informal settlements may include multiple locations where displaced people are sheltering within a given neighborhood.

The organizations sending aid to these areas in Türkiye include IOM, Turkish Red Crescent, ASAM, UNDP, UNICEF, UNFPA, UNHCR, UN Women, WFP, WHO, ILO, support to life, Ahbap Association, İHH, DFT (Development Foundation of Türkiye), WHH, Young Lives Foundation, Save the Children International, Türkiye Diyanet Foundation, metropolitan municipalities, local municipalities, and companies providing logistics services. These institutions and organizations abstracted with Organization in Hyperledger Fabric. These organizations connect to the blockchain network; they provide components such as peers (computers and services), chaincodes (smart contracts), and ordering service nodes. By incorporating these components into the network, organizations enlarge the network, increase the durability and security of the network, and ultimately contribute to creating an ideal chain structure.

Among the transactions that can be done in this channel are creating an aid request, updating the aid request, and querying the aid request. Saving the information beforehand regarding donations, is aimed at preventing any misuse or ill-usage of in-kind aid in the future (saving the attribute, amount, etc. of donations, even if they are changed later). In the query, there is a section that displays all the inventory information of the donated item as well as its location. How to perform spatial checks is explained in the "Geospatial Checks" section below.

The following information about donated items will be stored in the database receiving date, delivery date, receiving location (ADM1, ADM2, ADM3, latitude, longitude), delivery location (ADM1, ADM2, ADM3, shape: polygon), category, item name, status, brand, model, unit (amount), amount, price, condition, expire date. The visual version of this structure is shown in Figure 4.2. The information of the institutions or organizations that will be included in the network and the branches and warehouses where they accept in-kind aid will also be available in the network database. The data structure for this is also available in Figure 4.2. In addition, the data and data types that will be stored in the database where information regarding aid delivery targets (formal sites, informal sites, or homes of people in need) will be collected are shown in Figure 4.2. The geographical stations of an aid item are respectively in the order of branch of the organization to which the aid is delivered, AFAD / Institution / Organization warehouse, and settlement of the person in need. This order may change depending on which step the aid-sending process is started from (if it is started from the warehouse, the first step is not the organization branch).

3.4.2. Aids Provided to Migrants in Vulnerable Situations

This channel is dedicated to aid sent to migrants in vulnerable situations within the country. The humanitarian needs of these migrants in Türkiye are met through Presidency of Migration Management and through organizations operating within the country. The main organizations implementing these aids in Türkiye include IOM, Turkish Red Crescent, ASAM, support to life, İHH, WFP, HRDF, MUDEM, Save the Children International, UNICEF, UNFPA, Relief International, CARE International, GOAL, DRC, Global Communities, Solidarity Respect & Protect, Malteser International, World Vision International, ACTED, Doctors Worldwide Türkiye, metropolitan municipalities, local municipalities, and companies providing logistics services. These organizations connect to the blockchain network; they provide resources such as peers, chaincodes (smart contracts), and ordering service nodes. By incorporating these resources into the network, organizations enlarge the network, increase the durability and security of the network, and ultimately contribute to creating an ideal chain structure. The operations that can be performed in this channel are the same as in the previous two channels, and the Geospatial Checks section explains how to perform spatial checks for location tracking. The data of the donated items to be kept in the database are the same as in the previous two channels, and they are given with their data types in Figure 4.2. In addition, the data model for the buildings where the institutions or organizations that will provide the aid will store in-kind aid is also available in Figure 4.2.

3.4.3. Geospatial Checks

Smart contracts will be used to track donations transparently in terms of location. The location of donated items is tracked in stages: branch locations of organizations, organization warehouse locations, and locations of the beneficiaries. This ensures that both the donor and the organization can see if the donation is in the correct location. This location tracking step is envisioned to overcome the trust issue between the donor and the organization accepting the donation.

There are geographical stations in some steps of the workflow. Their actors and the duties during which these actors perform this are listed below. These are, "delivery acceptor/warehouse worker – ID code is generated for each aid", "carrier – transporting the items to the target area", and "distributor – local dispatcher takes aid to its destination".

3.4.3.1. In-kind aid ID generation

In these steps, the donor, who is the first link in the chain, delivers the in-kind aid to donate to the organization's staff. The staff records the donated items in a sort of inventory, assigns an ID number to each item, and takes photographs of them. This ID assignment and the photograph taken are proof that a new aid material has been delivered to a branch or warehouse of the organization, and since the buildings of the organizations are registered in the system, locational matching starts from the first step.

When aid items change hands, transactions are carried out with bilateral verification. While the delivery acceptor delivers the aid items to the carrier, the warehouse worker notifies the system when delivering them to the distributor. At the same time, when receiving these items, the carrier and distributor also notify the system. If either party does not confirm this delivery, the transaction is not valid.

3.4.3.2. Transporting the aid items to the target area

In this step, the company providing the logistics service, which receives the aid items from the staff of the organization that will provide the aid, can deliver the aid to three different places depending on the scenario. These are the warehouse of the charity organization, the formal site authority, or directly to the person in need. No matter which option the carrier delivers the aid items to, it will not be deemed to have been delivered unless the GPS of the vehicle used by the carrier falls within the polygon boundaries of any aid delivery target option. In other words, from a technical perspective, point-in polygon analysis will be performed in this section. After entering this polygon, the carrier must take various photographs of the area where the aid items are unloaded and inside the vehicle and upload them to the system to support verification.

Again, when aid items change hands, transactions are carried out with bilateral verification in this part. The carrier notifies the system when delivering the aid items to the warehouse worker or formal site authority. At the same time, when receiving these items, the warehouse worker or the formal site authority notifies the system. If either party does not confirm this delivery, the transaction is not valid. But there is an exception here. If the carrier will deliver the aid goods directly to the person in need, it is not mandatory for the person in need to verify this. The reason for this is that, especially immediately after a disaster, people may not have access to a device that can make such verification, or they may not be in the psychological state to deal with this task at that moment.

In addition, the route that the carrier will take after receiving the aid is predetermined. It provides locational tracking not only when the items are delivered, but also during the transportation process, by making locational notifications that the items are on the correct route in predetermined polygons between two points.

3.4.3.3. Delivering aid to the final destination

Since the distributors are the last actors of the aid organizations, they are expected to perform locational tracking themselves when they deliver the aid to the relevant places. Distributors deliver aid items either to the formal site authority or directly to the person in need. Just like the carrier, no matter which option the dispatcher delivers the aid to, it is not deemed to have delivered the aid before the GPS of the vehicle or phone the distributor uses enters the boundaries of the aid delivery target polygon. Again, in this section, point-in polygon analysis will be performed. In addition, various photographs should be taken in the area where the aid items were left and uploaded to the system to support verification.

Like in others, transactions are carried out with bilateral verification when aid items change hands. The distributor notifies the system when delivering aid items to the formal site authority or directly to the person in need. At the same time, when receiving these items, the formal site authority also notifies the system. If either party does not confirm this delivery, the transaction is not valid. However, the same exceptional situation applies if the items are to be delivered directly to the person in need.

4. DESIGN AND IMPLEMENTATION

This chapter explains the location-based smart contracts for aid delivery. Hyperledger Fabric is used as the blockchain framework. Go programming language was preferred for coding smart contracts and data structures. In the subsections, the network architecture, the data model, the capabilities of the actors and what they cannot do, the location-based smart contract application, and the discussion section for future work are mentioned. This thesis focuses on conceptual design of the location-based smart contracts and does not include user space applications which may provide easy to use graphical user interfaces and interactive maps.

4.1. Network Architecture in Hyperledger Fabric

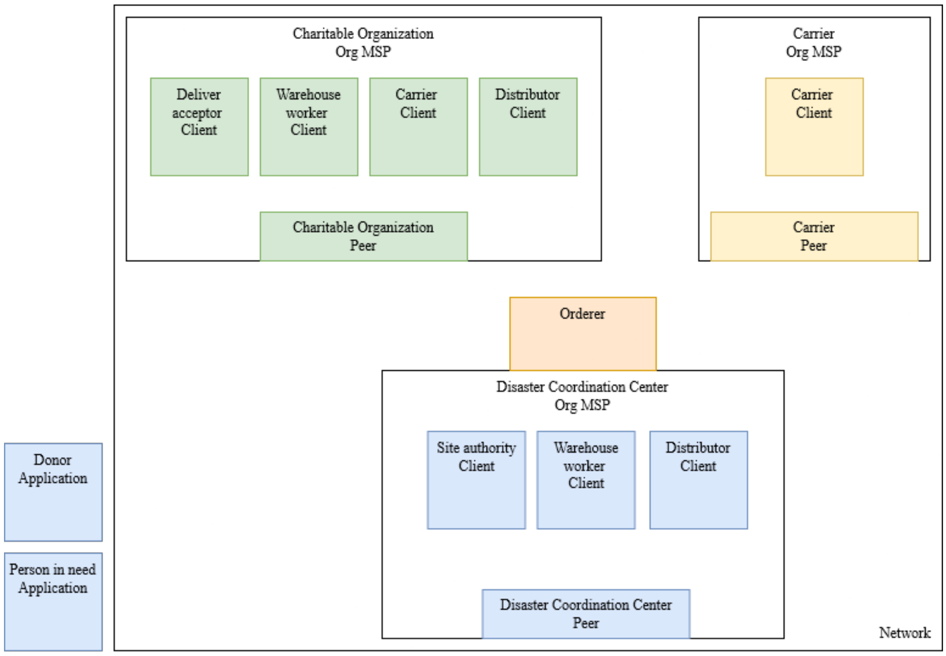


Figure 4.1. A network structure that includes orderer, peers, clients, and applications in their relevant organizations

The Hyperledger Fabric network, with its modular and flexible structure, supports various scenarios in a blockchain platform. This network design represents a complex structure involving the operations of a disaster coordination center and charitable organizations. In the following subsections detailed explanation of the network structure and its operations are listed in the context of Hyperledger constructs.

4.1.1. The Charitable Organization Org MSP

A charity that accepts and distributes donations are modeled by an organization in Hyperledger Fabric. It includes various client applications and a peer node. “The Deliver Acceptor Client” is used to receive and accept donations, ensuring that donations are correctly received and recorded. It may be a web based or mobile client to support the activities of the corresponding roles. “The Warehouse Worker Client” is used by warehouse personnel to manage donated materials and keep track of inventory. “The Carrier Client” is used when charitable organizations transport with their own logistics fleet. “The Distributor Client” manages the distribution of donated materials to those in need, tracking the distribution processes. “The Charitable Organization Peer” is the organization's peer node, responsible for verifying transactions and storing blockchain data, ensuring the reliability and integrity of the organization's transactions. It also includes shared smart contracts for location-based transaction checks.

4.1.2. The Carrier Org MSP

Logistics companies that transport donated items are modeled as organizations in Hyperledger Fabric. It includes a client application and a peer node. “The Carrier Client” manages the transportation operations, tracking the transportation processes and logistics operations. The client shall also interface with GPS tracking of vehicles within the organization. “The Carrier Peer” is the organization's peer node, responsible for recording and verifying transportation transactions on the blockchain. It also includes location-based smart contracts.

4.1.3. The Disaster Coordination Center Org MSP

In the Hyperledger framework, the Disaster Coordination Center is modeled as an organization that manages aid coordination in disaster areas. It includes various client applications and a peer node. “The Site Authority Client” is used by authorities in disaster areas to manage the distribution of aid and assess needs. “The Warehouse Worker Client” is used by warehouse personnel in disaster areas to manage the storage and distribution of aid. “The Disaster Coordination Center Peer” is the organization's peer node, responsible for verifying transactions and recording them on the blockchain. It also includes smart contracts.

The orderer plays a central role in the Hyperledger Fabric network by ordering transactions and creating blocks, which are then distributed to the nodes in the network. This component ensures that transactions are processed in the correct order and maintains the integrity of the blockchain data.

External applications play a crucial role in this network. Donors are not included in the network, but all these locational tracking operations are done for them. This application is available so that the donor can track the location of an aid item and view various information about the aid item. The person in need is also not included in the network. An external application is available so that they can view various information about incoming aid items. For external application development, each organization and also a central organization may provide a gateway interface to query the status of aid items given their ID.

All these operations are verified and recorded on the blockchain by the peer nodes of the respective organizations. This ensures that the operations are transparent, reliable, and traceable. Each peer node verifies the transactions of its own organization and contributes to the overall security of the blockchain network.

4.2. Data Model

Proposed database structure includes tracking of in-kind donations, buildings of organizations, location (information about the stops where the in-kind aid passes from the beginning to the last step is kept in this table.), and actor information. The database structure is presented in Figure 4.2. The data to be stored in this section and the explanations of properties are given below.

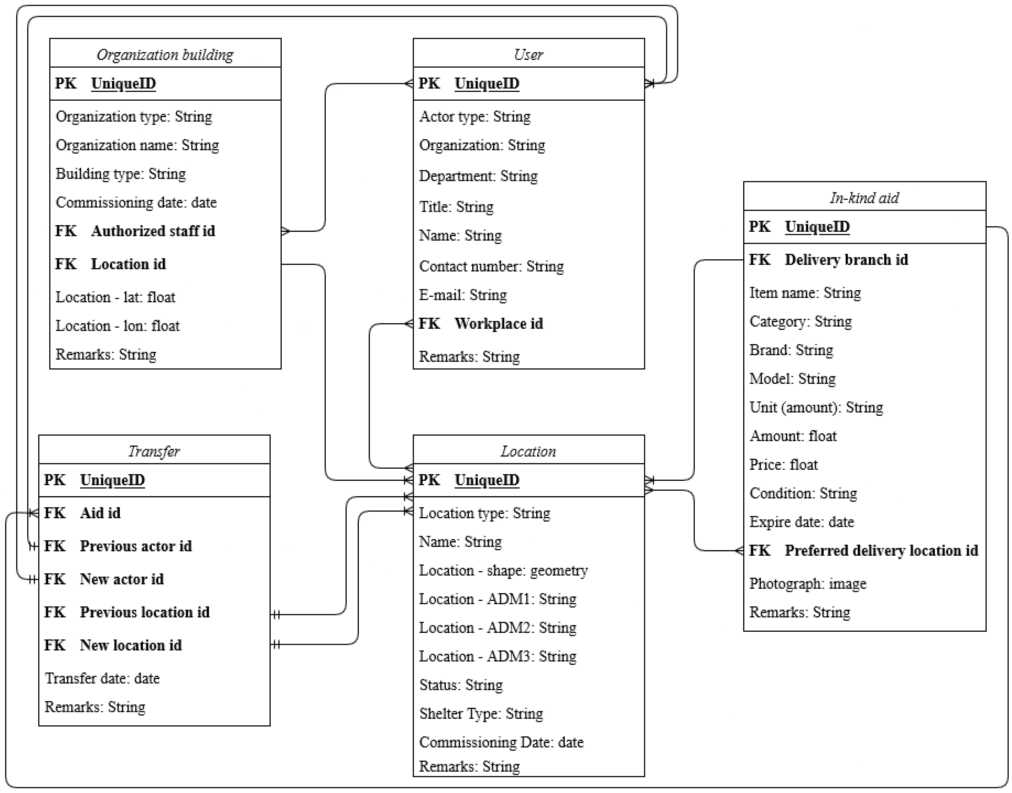


Figure 4.2. The data model of blockchain-based in-kind aid distribution network

The functions that each actor can perform within the design are explained in Table 4.1.

Table 4.1. Functions that actors can perform

Actor	Function	Description
Donor		
Delivery acceptor	Create in-kind aid	Creating in-kind aid item brought to the branch of the organization and entering inventory information.
	Update in-kind aid	Updating in-kind aid item brought to the branch of the organization.
	Update actor	Updating which actor currently has the in-kind aid.
	Update location	Updating the location of in-kind aid.
	Add photo	Supporting photographs to show that the items have been delivered to the correct location.
	Create control polygons	Creating control polygons for various locational validations.
	Check carrier	Checking which carrier provides service at which locations before delivering the items to the carrier.
Carrier	Update actor	Updating which actor currently has the in-kind aid.

Actor	Function	Description
	Check route control points	The carrier must follow a predetermined route while delivering the items and must stop at checkpoints on this route and report to the network that they are following the correct route. Technically, point in polygon analysis is performed in this function.
	Check warehouse	If the carrier is going to deliver the items to the warehouse of the organization, check that it is delivered to the correct warehouse. Technically, point in polygon analysis is performed in this function.
	Check delivery area	If the carrier is going to deliver the items to the area of need, check that it is delivered to the correct area of need. Technically, point in polygon analysis is performed in this function.
	Add photo	Supporting photographs to show that the items have been delivered to the correct location.
Warehouse worker	Create in-kind aid	Creating in-kind aid item brought to the branch of the organization and entering inventory information.
	Update in-kind aid	Updating in-kind aid item brought to the branch of the organization.
	Update actor	Updating which actor currently has the in-kind aid.
	Update location	Updating the location of in-kind aid.
	Add photo	Supporting photographs to show that the items have been delivered to the correct location.
	Create control polygons	Creating control polygons for various locational validations.
	Check distributor	Checking which distributor provides service at which locations before delivering the items to the distributor.
Distributor	Update actor	Updating which actor currently has the in-kind aid.
	Update location	Updating the location of in-kind aid.
	Add photo	Supporting photographs to show that the items have been delivered to the correct location.
	Check delivery area	If the distributor is going to deliver the items to the area of need, check that it is delivered to the correct area of need. Technically, point in polygon analysis is performed in this function.
	Check site authority	If the items are to be delivered to the site authority, it is checked beforehand whether they are delivered to the correct site authority.
Site authority	Update actor	Updating which actor currently has the in-kind aid.

Actor	Function	Description
	Add photo	Supporting photographs to show that the items have been delivered to the correct location.
Person in need		
Common*	Get information/location	Querying in-kind aid information / monitoring its location

*Function that can be performed by all actors.

4.2.1. In-kind Aid Table

This table is created to track non-monetary assistance such as hygiene, food, medical supplies, items needed for the continuity of shelter, and other basic items. Each record in this table contains detailed information about aid, including the type, amount, status, location, and parties involved in providing and receiving aid. By recording in-kind donations, each item of aid is recorded, reducing waste and preventing fraud.

```go
import (
    "time"
)

type InKindAid struct {
    ID                          string     `json:"id"`
    DeliveryBranchID            string     `json:"deliveryBranchID"`
    ItemName                    string     `json:"itemName"`
    Category                    string     `json:"category"`
    Brand                       string     `json:"brand"`
    Model                       string     `json:"model"`
    Unit                        string     `json:"unit"`
    Amount                      float64    `json:"amount"`
    Price                       float64    `json:"price"`
    Condition                   string     `json:"condition"`
    ExpireDate                  time.Time  `json:"expireDate"`
    PreferredDeliveryLocationID string     `json:"preferred_delivery_location_id"`
    Photograph                  string     `json:"photograph"` // base64 encoded string
    Remarks                     string     `json:"remarks"`
}
```

The decription of each attribute are as follows:

- ID: Unique identifier for the aid record.
- ItemName: Name of donated item.
- Category: Category of the aid (e.g., food, clothing).
- Brand: Brand of the aid.
- Model: Model of the aid.

- Unit: Unit of measure for the aid (e.g., kg, liters).
- Amount: Quantity of the aid.
- Price: Total value of the aid.
- Condition: Condition of the aid (e.g., new, used).
- ExpireDate: Expiry date of the aid (for things like food).
- PreferredDeliveryLocationID: The area where the donor prefers aid to be delivered.
- Photograph: Photograph of the aid (base64 encoded string).
- Remarks: Additional remarks or notes.

4.2.2. Location Table

Geographic information plays a critical role in humanitarian aid management. This table stores information about specific geographic areas; These areas can represent anything from refugee camps to areas affected by natural disasters. Apart from a camp structure, it can also represent the home of any single family in need. By keeping detailed stores about these areas, including the number of people involved, the type of shelter and their geographical location, aid distribution to targeted vulnerable areas can be ensured efficiently. Another important aspect of this table is that it stores the polygons required for point in polygon analysis for proof of location.

In addition, the carrier must verify that it is going on the correct route spatially, and it is expected to follow this by verifying from predefined polygon areas and reporting it to the system.

```
type Location struct {
    ID               string      `json:"id"`
    LocationType     string      `json:"locationType"`
    Name             string      `json:"name"`
    LocationShape    orb.Polygon `json:"locationShape"` // geojson format
    LocationADM1     string      `json:"locationADM1"`
    LocationADM2     string      `json:"locationADM2"`
    LocationADM3     string      `json:"locationADM3"`
    Status           string      `json:"status"`
    ShelterType      string      `json:"shelterType"`
    CommissioningDate time.Time  `json:"commissioningDate"`
    Remarks          string      `json:"remarks"`
}
```

The description of each element in this code is as follows:

- ID: Unique identifier for the polygon.

- LocationType: Type of polygon (e.g., delivery stop, organization warehouse, final destination).
- Name: A naming of the area.
- LocationShape: Geographical shape of the polygon (in GeoJSON format).
- LocationADM1: Administrative region 1 where the polygon is located (the equivalent in Türkiye is a province.).
- LocationADM2: Administrative region 2 where the polygon is located (the equivalent in Türkiye is a sub-province.).
- LocationADM3: Administrative region 3 where the polygon is located (the equivalent in Türkiye is a neighborhood.).
- Status: Status of the location (e.g., active, inactive).
- ShelterType: If it is a need area, it is the information about its type (e.g., tent city, container city, home).
- CommissioningDate: Date when the area/building was commissioned.
- Remarks: Additional remarks or notes.

4.2.3. User Table

All users involved in humanitarian aid processes from the beginning to the end are stored under this table structure. This table shows the types of users, it includes information about the organization and contact information.

```
type User struct {
   ID           string `json:"id"`
   ActorType     string `json:"actorType"`
   Organization  string `json:"organization"`
   Department    string `json:"department"`
   Title         string `json:"title"`
   Name          string `json:"name"`
   ContactNumber string `json:"contactNumber"`
   Email         string `json:"email"`
   WorkplaceID   string `json:"workplaceID"`
   Remarks       string `json:"remarks"`
}
```

The description of each element in this code is as follows:

- ID: Unique identifier for the user.
- ActorType: Type of actor (e.g., warehouse worker, carrier).
- Organization: Name of the organization.

- Department: Name of the department in the organization of the user.
- Title: Name of the title in the organization of the user.
- Name: Name of the user.
- ContactNumber: Phone number of the user.
- ContactEmail: E-mail of the user.
- WorkplaceID: Location of the user.
- Remarks: Additional remarks or notes.

4.2.4. Organization Building Table

The storage of humanitarian aid items is generally hosted by warehouses, offices, etc. structures of relevant institutions and organizations providing services in this field. The organization building table stores basic information such as the facilities and geographical location of these structures. This information is critical for logistics planning and resource allocation, ensuring efficient use of existing infrastructure. In addition, these buildings are also used for proof of location, due to the fact that aid items indicate, for instance, that they were in a warehouse of the organization at that moment.

```
type Organization_Building struct {
   ID                 string     `json:"id"`
   OrganizationType   string     `json:"organizationType"`
   OrganizationName   string     `json:"organizationName"`
   BuildingType       string     `json:"buildingType"`
   AuthorizedStaffID  string     `json:"authorizedStaffID"`
   LocationID         string     `json:"locationID"`
   LocationLat        float64    `json:"location_lat"`
   LocationLon        float64    `json:"location_lon"`
   Remarks            string     `json:"remarks"`
}
```

The description of each element in this code is as follows:

- ID: Unique identifier for the building.
- OrganizationType: Type of the organization (e.g., INGO, NGO, government organization).
- OrganizationName: Name of the organization.
- BuildingType: Type of the building (e.g., office, warehouse).
- AuthorizedStaffID: Information about the authorized staff for the building.
- LocationID: Information about the location of the building.

- LocationLat: Latitude of the building.
- LocationLon: Longitude of the building.
- Remarks: Additional remarks or notes.

4.2.5. Transfer Table

This data structure was created to track both the location of the aid items and which actor has them.

```
// Transfer represents the transfer details of the aid.
type Transfer struct {
   ID                  string       `json:"uniqueID"`
   AidID               int          `json:"aidID"`
   PreviousActorID     int          `json:"previousActorID"`
   NewActorID          int          `json:"newActorID"`
   PreviousLocationID  int          `json:"previousLocationID"`
   NewLocationID       int          `json:"newLocationID"`
   TransferDate        time.Time    `json:"transferDate"`
   Remarks             string       `json:"remarks"`
}
```

The description of each element in this code is as follows:

- ID: Unique identifier for the building.
- PreviousActorID: The actor who previously had the aid item.
- NewActorID: The actor who currently has the aid item.
- PreviousLocationID: The location where the aid item was previously located.
- NewLocationID: The location where the aid item is currently located.
- TransferDate: The date the transfer occurred.
- Remarks: Additional remarks or notes.

4.2.6. Working Area Table

The Working Area structure contains the administrative divisions (ADM1, ADM2, ADM3) where a user works or is responsible. With this structure, a definition is made for different administrative divisions and it becomes easier to track the working areas of specific users.

```
type Working_Area struct {
   ID      string `json:"id"`
   ADM1    string `json:"adm1"`
   ADM2    string `json:"adm2"`
   ADM3    string `json:"adm3"`
   Remarks string `json:"remarks"`
```

```
}
```
The description of each element in this code is as follows:

- ID: Unique identifier for the building.
- ADM1: Administrative region 1 where the polygon is located (the equivalent in Türkiye is a province.).
- ADM2: Administrative region 2 where the polygon is located (the equivalent in Türkiye is a sub-province.).
- ADM3: Administrative region 3 where the polygon is located (the equivalent in Türkiye is a neighborhood.).
- Remarks: Additional remarks or notes.

4.3. Abilities and Restrictions of Actors

In a workflow scenario where all stakeholders are involved, the actors are donor, delivery acceptor, carrier, warehouse worker, distributor, site authority, and person in need. This study focuses more on the use of location-based checks within the blockchain network, however to support in-kind aid delivery additional business rules shall also be implemented in smart contracts. This section describes the roles of the various actors involved in the aid delivery process in Hyperledger Fabric.

4.3.1. Donor

Donors are the initiating actors in the process of delivering aid to disadvantaged groups. Their roles are as follows:

- Can view all data of donated in-kind aid within the "InKindAid" structure.
- Can view all data within the "Transfer" structure.

4.3.2. Delivery Acceptor

Delivery acceptors are responsible for receiving aid from donors and properly transferring it to carriers. Their roles are as follows:

- Can view, edit, and query all data of donated in-kind aid within the "InKindAid" structure.
- Can view, edit, and query all data within the "Organization_Building" structure.
- Can view, edit, and query all data within the "Location" structure.

- Can view, edit, and query all data within the "Transfer" structure. When aid items change hands, transactions are carried out with bilateral verification. The delivery acceptor notifies the system about this when delivering aid items to the carrier. At the same time, when receiving these items, the carrier also notifies the system about this. If one of the parties does not approve this delivery, the transaction is not considered valid.
- Can view, edit, and query all data within the "User" structure.
- Can view, edit, and query all data within the "Working_Area" structure.

4.3.3. Carrier

Carriers are responsible for transporting aid. Their roles are as follows:

- "ActorID", "DeliveryDate", "Photograph", "Remarks" data in the "InKindAid" structure of the donated in-kind aid can be viewed, edited, and queried. “ReceivingDate”, “Item name”, “Category”, “Status”, “Brand”, “Model”, “Unit”, “Amount”, “Price”, “Condition”, “ExpireDate”, “ReceivingBuildingId”, “PreferredDeliveryLocationId”, “DeliveryLocationId” ” can only be viewed and queried by this actor.
- Can view and query all data in the “Location” structure. They can only edit items for which "LocationType" is route control.
- Can view and query all data in the "Organization_Building" structure.
- Can view, edit, and query all data within the "Transfer" structure. When aid items change hands, transactions are carried out with bilateral verification. The carrier notifies the system about this when delivering aid items to the warehouse worker or the site authority. At the same time, when receiving these items, the warehouse worker or the site authority also notifies the system about this. If one of the parties does not approve this delivery, the transaction is not considered valid but there is an exception. If the carrier will deliver the aid items directly to the person in need, the person in need is not expected to verify this. It is optional for the person in need to notify the system when receiving these items. The reason for this is that, especially immediately after a disaster, people may not have access to a device that can make such verification, or they may not be in the mental state to deal with this task at that moment.

4.3.4. Warehouse Worker

Warehouse workers manage the process of storing aid items. Their roles are as follows:

- Can view, edit, and query all data of donated in-kind aid within the "InKindAid" structure.
- Can view, edit, and query all data within the "Location" structure.
- Can view, edit, and query all data within the "Organization_Building" structure.
- Can view, edit, and query all data within the "Transfer" structure. When aid items change hands, transactions are carried out with bilateral verification. The warehouse worker notifies the system about this when delivering aid items to the distributor. At the same time, when receiving these items, the distributor also notifies the system about this. If one of the parties does not approve this delivery, the transaction is not considered valid.
- Can view, edit, and query all data within the "User" structure.

4.3.5. Distributor

Distributors deliver aid to the site authorities or directly to the person in need. Their roles are as follows:

- Can view and query all data of donated in-kind aid within the "InKindAid" structure.
- Can view, edit, and query all data within the "Location" structure.
- Can view, edit, and query all data within the "Organization_Building" structure.
- Can view, edit, and query all data within the "Transfer" structure. When aid items change hands, transactions are carried out with bilateral verification. The distributor notifies the system about this when delivering aid items to the site authority or the person in need. At the same time, when receiving these items, the site authority also notifies the system about this. If one of the parties does not approve this delivery, the transaction is not considered valid but there is an exception. If the distributor will deliver the aid items directly to the person in need, the person in need is not expected to verify this. It is optional for the person in need to notify the system when receiving these items. The reason for this is that, especially immediately after a disaster, people may not have access to a device that can make such verification, or they may not be in the mental state to deal with this task at that moment.

- Can view and query all data within in the "User" structure.
- Can view and query all data within the "Working_Area" structure.

4.3.6. Site Authority

Site authorities are responsible for the fair distribution of aid in formal sites (container city, tent city, refugee camp, irregular settlement not affiliated with a formal organization, home, etc.). Their roles are as follows:

- Can view and query all data of donated in-kind aid within the "InKindAid" structure. Can only edit the condition of in-kind aid items.
- Can view, edit, and query all data within the "Location" structure.
- Can view and query all data within the "Organization_Building" structure.
- Can view, edit, and query all data within the "Transfer" structure. When aid items change hands, transactions are carried out with unilateral verification. The site authority notifies the system about this when delivering aid items to the person in need. It is optional for the person in need to notify the system when receiving these items. The reason for this is that, especially immediately after a disaster, people may not have access to a device that can make such verification, or they may not be in the mental state to deal with this task at that moment.

4.3.7. Person in Need

Person in need is the ultimate goal of aid. Their roles are as follows:

- Can view the “DeliveryDate”, “Item name”, “Category”, “Brand”, “Model”, “ExpireDate”, “Photograph” data within the "InKindAid" structure of the donated in-kind aid.
- Can view the transfer information of the aid items that have reached or will reach them.

4.4. Location-Based Smart Contract Implementation

In this section, the functions used in the smart contract structure are discussed. In addition, queries and invokes are explained in a story-like manner in the code written at a basic level for application demonstration.

Besides the CRUD applications of the structures defined in the data model from the previous sections within the smart contract, a function also performs point in polygon analysis.

4.4.1. Create Functions

The in-kind aid structure is given as an example to explain the “create” functions. This function contains all the steps required to add a new "InKindAid" asset, especially an aid item, to a Hyperledger Fabric network. First, it checks if the asset already exists, then a new asset is created, and this asset is converted to JSON format and finally saved to the blockchain network.

```
func (s *SmartContract) CreateInkindAid(ctx contractapi.TransactionContextInterface,
  id string, DeliveryBranchID string, ItemName string, Category string, Brand string, Model string,
  Unit string, Amount string, Price string, Condition string, ExpireDate string,
  PrefferedDeliveryLocationId string, CurrentLocationID string, CurrentActorID string, Photograph
string, Remarks string) error {

  // Check if the InKindAid asset already exists using the provided ID
  exists, err := s.InkindAidExists(ctx, id)
  if err != nil {
    // Return an error if there was an issue checking existence
    return err
  }
  if exists {
    // If the asset already exists, return an error indicating duplication
    return fmt.Errorf("the aid %s already exists", id)
  }

  // Create a new instance of the InKindAid struct with the provided data
  aid := InKindAid{
    ID:                          id,
    DeliveryBranchID:           DeliveryBranchID,
    ItemName:                   ItemName,
    Category:                   Category,
    Brand:                      Brand,
    Model:                      Model,
    Unit:                       Unit,
    Amount:                     Amount,
    Price:                      Price,
    Condition:                  Condition,
    ExpireDate:                 ExpireDate,
    PreferredDeliveryLocationID: PrefferedDeliveryLocationId,
    CurrentLocationID:          CurrentLocationID,
    CurrentActorID:             CurrentActorID,
    Photograph:                 Photograph,
```

```
        Remarks:                Remarks,
    }

    // Convert the InKindAid struct into JSON format
    assetJSON, err := json.Marshal(aid)
    if err != nil {
        // Return an error if there was an issue with JSON marshalling
        return err
    }

    // Store the JSON-formatted asset in the blockchain state using the provided ID
    return ctx.GetStub().PutState(id, assetJSON)
}
```

4.4.2. Read Functions

The location structure is given as an example to explain the “read” functions. This function reads a "Location" asset in the blockchain network. The function first checks whether the ID is in the correct format, and then reads the data corresponding to the specified ID from the blockchain. After reading the data, it converts the data in JSON format to the Location structure and returns it. If an error occurs at any stage, this error is returned. This process ensures that entities are read correctly in blockchain-based applications.

```
func (s *SmartContract) ReadLocation(ctx contractapi.TransactionContextInterface, id string)
(*Location, error) {

    // Check if the provided ID starts with the required prefix "Loc-"
    if !strings.HasPrefix(id, LOCATION_ID_PREFIX) {
        // If the ID does not start with the prefix, return an error indicating the invalid format
        return nil, fmt.Errorf("the location Ids must start with Loc- already exists")
    }

    // Retrieve the location data from the world state using the provided ID
    locJSON, err := ctx.GetStub().GetState(id)
    if err != nil {
        // If there is an error reading from the world state, return an error with details
        return nil, fmt.Errorf("failed to read from world state. %s", err.Error())
    }
    if locJSON == nil {
        // If no data is found for the provided ID, return an error indicating that the asset does not exist
        return nil, fmt.Errorf("the asset %s does not exist", id)
    }

    // Declare a variable to hold the unmarshaled Location struct
    var location Location
    // Unmarshal the JSON data into the Location struct
```

```
    err = json.Unmarshal(locJSON, &location)
    if err != nil {
        // If there is an error unmarshaling the JSON data, return the error
        return nil, err
    }

    // Return the Location struct and a nil error indicating success
    return &location, nil
}
```

4.4.3. Update Functions

The user structure is given as an example to explain the “update” functions. This function first checks if a user with the specified ID exists. This step ensures that only existing users' department information is updated. If the user does not exist, the function returns an error. The function creates a new user structure with the specified ID and department information if the user exists. This structure overwrites the old record. The updated user structure is converted to JSON format and stored in the blockchain.

```
func (s *SmartContract) UpdateDepartment(ctx contractapi.TransactionContextInterface, id,
department string) error {
    // Check if the user with the given ID exists in the world state
    exists, err := s.UserExists(ctx, id)
    if err != nil {
        // If there is an error checking for the existence of the user, return the error
        return err
    }
    if !exists {
        // If the user does not exist, return an error indicating that the user cannot be updated
        return fmt.Errorf("the user %s does not exist", id)
    }

    // Create a new User struct with the provided ID and department
    // This will overwrite the original user record with the updated department information
    user := User{
        ID:         id,
        Department: department,
    }

    // Marshal the User struct to JSON format to prepare it for saving
    userJSON, err := json.Marshal(user)
    if err != nil {
        // If there is an error marshalling the User struct to JSON, return the error
        return err
    }

    // Write the updated user data back to the world state with the same ID
```

```
    return ctx.GetStub().PutState(id, userJSON)
}
```

4.4.4. Delete Functions

The in-kind aid structure is given as an example to explain the “delete” functions. The process checks if an "InkindAid" asset with the specified ID exists and if the asset exists, deletes it from the blockchain network.

```
func (s *SmartContract) DeleteInkindAid(ctx contractapi.TransactionContextInterface, id string)
error {
    // Check if the InkindAid asset with the given ID exists in the world state
    exists, err := s.InkindAidExists(ctx, id)
    if err != nil {
        // If there is an error checking for the existence of the asset, return the error
        return err
    }
    if !exists {
        // If the asset does not exist, return an error indicating that the asset cannot be deleted
        return fmt.Errorf("the aid %s does not exist", id)
    }

    // If the asset exists, delete it from the world state
    return ctx.GetStub().DelState(id)
}
```

4.4.5. Point in Polygon Function

This function checks if the coordinates of a given point are inside a polygon stored in the blockchain. The polygon data in this function is stored inside the "Location" struct.

An instance of the use of function through the actors in this thesis is that a carrier needs to pass through certain polygons while making a delivery. This way, it can be understood that it is proceeding on the correct route. Another instance is the control that the distributor performs to deliver the aid to the correct area, in this part, it sends its location to the blockchain network and receives the answer whether it is in the correct area or not.

```
// PointInPolygon checks if a given point is inside a polygon and returns a message accordingly
// The function takes in coordinates of a point (pointX, pointY) and a polygon's ID (polygonID) to fetch
the polygon data from the ledger.
// It will determine if the point lies within the polygon and return a corresponding message.
func (s *SmartContract) PointInPolygon(ctx contractapi.TransactionContextInterface, pointX
float64, pointY float64, polygonID string) (bool, string, error) {

    // Retrieve the polygon (location) from the ledger using the polygonID.
    // The ctx.GetStub().GetState function fetches the location data from the ledger where polygonID is
used as the key.
```

```
    // If the ledger access fails, an error is returned.
    locationAsBytes, err := ctx.GetStub().GetState(polygonID)
    if err != nil {
        return false, "", fmt.Errorf("failed to read location: %v", err)
    }

    // If the ledger does not contain any data with the given polygonID, return an error message indicating
that the location was not found.
    if locationAsBytes == nil {
        return false, "", fmt.Errorf("location not found: %s", polygonID)
    }

    // Define a variable of type Location to hold the unmarshaled polygon data.
    var location Location

    // Unmarshal the retrieved location JSON data (locationAsBytes) into the 'location' struct.
    // This converts the stored JSON format into the Go data structure.
    // If the unmarshaling process fails, return an error indicating the issue.
    err = json.Unmarshal(locationAsBytes, &location)
    if err != nil {
        return false, "", fmt.Errorf("failed to unmarshal location: %v", err)
    }

    // Create a point using the input coordinates pointX and pointY. This point will be checked if it lies
within the polygon.
    // orb.Point is used here to represent the point.
    point := orb.Point{pointX, pointY}

    // Check if the point is inside the polygon by using the planar.PolygonContains function from the
orb/planar package.
    // The function checks if the point lies within the polygon (location.LocationShape).
    inside := planar.PolygonContains(location.LocationShape, point)

    // Based on the result of the check (whether the point is inside the polygon or not), return an
appropriate message.
    if inside {
        return true, fmt.Sprintf("Point (%.5f, %.5f) is INSIDE the polygon with ID: %s", pointX, pointY,
polygonID),  inside
    } else {
        return false, fmt.Sprintf("Point (%.5f, %.5f) is OUTSIDE the polygon with ID: %s. Please go to the
polygon to complete the transfer process!", pointX, pointY, polygonID), inside
    }
}
```

4.4.6. Transfer Function

This function allows the transfer of in-kind aid assets. Its main purpose is to transfer in-kind aid from its previous owner to its new owner, updating both location and actor information during this transfer.

```
// TransferInKindAid handles the transfer of an InKindAid asset to a new actor and/or new location.
func (s *SmartContract) TransferAid(ctx contractapi.TransactionContextInterface, aidID string,
previousActorID string, newActorID string, previousLocationID string, newLocationID string) error {

   // Retrieve the InKindAid from the ledger using the aidID
   inKindAidAsBytes, err := ctx.GetStub().GetState(aidID)
   if err != nil {
      return fmt.Errorf("failed to read InKindAid: %v", err)
   }
   if inKindAidAsBytes == nil {
      return fmt.Errorf("InKindAid not found: %s", aidID)
   }

   // Retrieve the existing Transfer record using the aidID
   transferID := "Trn-" + aidID
   transferAsBytes, err := ctx.GetStub().GetState(transferID)
   if err != nil {
      return fmt.Errorf("failed to read Transfer record: %v", err)
   }

   var transfer Transfer
   if transferAsBytes != nil {
      err = json.Unmarshal(transferAsBytes, &transfer)
      if err != nil {
         return fmt.Errorf("failed to unmarshal Transfer: %v", err)
      }
   } else {
      transfer.ID = transferID
      transfer.AidID = aidID
   }

   // Update the PreviousActorID and NewActorID fields
   transfer.PreviousActorID = previousActorID
   transfer.NewActorID = newActorID

   // Update the PreviousLocationID and NewLocationID fields
   transfer.PreviousLocationID = previousLocationID
   transfer.NewLocationID = newLocationID

   // Marshal the updated Transfer back to JSON
   transferAsBytes, err = json.Marshal(transfer)
   if err != nil {
```

```
        return fmt.Errorf("failed to marshal Transfer: %v", err)
    }

    // Save the updated Transfer record back to the ledger
    err = ctx.GetStub().PutState(transfer.ID, transferAsBytes)
    if err != nil {
        return fmt.Errorf("failed to update Transfer record: %v", err)
    }

    return nil
}
```

4.4.7. Working Area Check Function

Before delivering the aid item from the delivery acceptor to the carrier, the warehouse worker to the distributor, and the distributor to the site authority, they must check that the aid will be delivered to the right person. The right person varies according to the type of actor and generally means an employee who is assigned to the relevant region or can go to the relevant region. The function examined under this heading was created to detect the right person.

This function runs a query on CouchDB to find users related to a specific "workplaceID". The query results are processed using an iterator, the JSON data is converted to a "User" structure, and these users are returned as a slice. If any error occurs during the process or no user is found related to the specified "workplaceID", the function returns an error message. This process allows users to be filtered according to their working area.

```
func (s *SmartContract) WorkingAreaCheck(ctx contractapi.TransactionContextInterface,
workplaceID string) ([]*User, error) {
    // Construct a CouchDB query to find users based on the workplaceID
    queryString := fmt.Sprintf(`{"selector":{"workplaceID":"%s"}}`, workplaceID)

    // Execute the query and get an iterator for the results
    resultsIterator, err := ctx.GetStub().GetQueryResult(queryString)
    if err != nil {
        // If there is an error executing the query, return an error message
        return nil, fmt.Errorf("failed to get user details: %v", err)
    }
    // Ensure that the results iterator is closed after the function returns
    defer resultsIterator.Close()

    // Initialize an empty slice to store the users
    var users []*User
    // Iterate through the query results
    for resultsIterator.HasNext() {
```

```
        // Get the next item from the iterator
        queryResponse, err := resultsIterator.Next()
        if err != nil {
            // If there is an error getting the next item, return an error message
            return nil, fmt.Errorf("failed to get next user: %v", err)
        }

        // Unmarshal the JSON data into a User struct
        var user User
        err = json.Unmarshal(queryResponse.Value, &user)
        if err != nil {
            // If there is an error unmarshaling the data, return an error message
            return nil, fmt.Errorf("failed to unmarshal user: %v", err)
        }

        // Append the user to the slice of users
        users = append(users, &user)
    }

    // If no users were found, return an error indicating that no users were found for the given
workplaceID
    if len(users) == 0 {
        return nil, fmt.Errorf("no users found with workplaceID: %s", workplaceID)
    }

    // Return the slice of users and a nil error indicating success
    return users, nil
}
```

4.5. Sample Scenario

In this section, the functions within the Hyperledger Fabric blockchain network are explained practically by telling a story. This story represents an ideal aid delivery scenario and does not cover situations where error messages should be returned. All characters and organizations used in the sample scenario are randomly generated and do not represent real persons or institutions/organizations.

Türkiye starts the day with a severe earthquake felt in Hatay and nearby provinces. Meanwhile, Ayşe lives in Ankara and feels deeply saddened by the horrific images of Hatay, the province where she was born and raised, reflected in the media. She wants to help immediately as much as she can and for this purpose. She buys diapers by referring to the needs list announced by the Disaster Coordination Center and donates these aid items to the X Association to be delivered to the earthquake region. Ayşe requests Bilge Erduran (delivery acceptor), a volunteer at the X Association to whom she delivered the donation, to send the aid items to Hatay if possible since she was born and raised in Hatay and has emotional ties to Hatay (there are also other provinces affected by the earthquake such as Adıyaman, Kahramanmaraş, Malatya and Gaziantep). Bilge considers this when entering the information about the aid items into the blockchain network and selects Hatay (Loc-2) as the preferred delivery location for the aid items. The elements written in the "Args" are the attributes of the diaper.

```
ubuntu@aid-transfer-dev:~/aid-transfer/contracts/chaincode-external$
 peer chaincode invoke -C formal -n asset-transfer -c '{"function":"
CreateInkindAid", "Args":["Aid-2", "Bld-1", "Diaper", "Hygiene", "Di
aper_Brand", "Maxi", "Piece", "1000", "2790", "New", "n/a", "Loc-2",
 "molfix_diaper_100.png", "n/a"]}' | jq
```

Before delivering these aid items to a carrier, Bilge checks which carriers are working in the region by querying the network (Wor-1: Hatay), as a result of this check, she decides to carry out this transportation with Y Logistics company.

```
ubuntu@aid-transfer-dev:~/aid-transfer/contracts/chaincode-external$
 peer chaincode query -C formal -n asset-transfer -c '{"function":"W
orkingAreaCheck", "Args":["Wor-1"]}' | jq
[
  {
    "id": "Usr-2",
    "actorType": "Carrier",
    "organization": "Y Logistics",
    "department": "Land Transportation",
    "title": "Truck Driver",
    "name": "Ümit Zengin",
    "contactNumber": "90 3121231234",
    "email": "uzengin@ylogistic.com",
    "workplaceID": "Wor-1",
    "remarks": ""
  },
```

Bilge delivers the aid items to Ümit Zengin, a carrier working in the region (Usr-1: Bilge Erduran, Usr-2: Ümit Zengin, Loc-1: Ankara Branch of X Association, Loc-3: Y Logistics Ankara - 1).

```
ubuntu@aid-transfer-dev:~/aid-transfer/contracts/chaincode-external$
 peer chaincode invoke -C formal -n asset-transfer -c '{"function":"
TransferAid", "Args":["Aid-2", "Usr-1", "Usr-2","Loc-1","Loc-3"]}' |
 jq
2024-09-04 01:39:07.795 +03 0001 INFO [chaincodeCmd] InitCmdFactory
-> Retrieved channel (formal) orderer endpoint: orderer-api.127-0-0-
1.nip.io:8080
2024-09-04 01:39:07.868 +03 0002 INFO [chaincodeCmd] chaincodeInvoke
OrQuery -> Chaincode invoke successful. result: status:200
```

While Ümit is taking the aid items from Ankara to Hatay, he needs to stop by certain route checkpoints to show that he is on the right route. These are Niğde Toll Booths and

Derinkuyu Checkpoint (The information and polygon coordinates of these two locations are below.), respectively.

```
// Define the locations
locations := []Location{
...
{ID: "Loc-2", LocationType: "Formal Site", Name: "Büyükdalyan Temporary Accommodation Center",
LocationADM1: "Hatay", LocationADM2: "Antakya", LocationADM3: "Buyukdalyan Mahallesi",
Status: "Active", ShelterType: "Container"},
{ID: "Loc-4", LocationType: "Organization Warehouse", Name: "Antakya Warehouse", LocationADM1:
"Hatay", LocationADM2: "Iskenderun", LocationADM3: "Askarbeyli Mahallesi", Status: "Active"},
{ID: "Loc-5", LocationType: "Route Control", Name: "Niğde Toll Booths", LocationADM1: "Nigde",
LocationADM2: "Nigde Merkez", LocationADM3: "Pinarcik Koyu", Status: "Active"},
{ID: "Loc-6", LocationType: "Route Control", Name: "Derinkuyu Checkpoint", LocationADM1:
"Nevsehir", LocationADM2: "Derinkuyu", LocationADM3: "Suvermez Koyu", Status: "Active"},
}

// Define the polygons (corresponding to each location)
polygons := []orb.Polygon{
...
{{{36.298093828460985, 36.194366040006294}, {36.2969270118326, 36.19796959563141},
{36.30038749381797, 36.194769566572994}, {36.300034794555216, 36.19898141124972}}},
{{{36.34810833459535, 36.209319337654414}, {36.346861694601884, 36.21160568832594},
{36.34475653141238, 36.21083949996912}, {36.345275618023855, 36.207774746672456}}},
{{{38.145268589723344, 34.715049739878154}, {38.14538219930656, 34.719646173930684},
{38.139784134891556, 34.7155619139583}, {38.13966018764583, 34.718280376383646}}},
{{{38.3438250737882, 34.55701462675088}, {38.34276961195441, 34.5810949902165},
{38.334399639346785, 34.57480256153113}, {38.33626120800884, 34.56227357368168}}},
}
```

When he arrives at these points, he reports his location to the network.

```
ubuntu@aid-transfer-dev:~/aid-transfer/contracts/chaincode-external$
 peer chaincode query -C formal -n asset-transfer -c '{"function":"P
ointInPolygon", "Args":["38.14260513903852", "34.71730695748988", "L
oc-5"]}'
Point (38.14261, 34.71731) is INSIDE the polygon with ID: Loc-5
ubuntu@aid-transfer-dev:~/aid-transfer/contracts/chaincode-external$
 peer chaincode query -C formal -n asset-transfer -c '{"function":"P
ointInPolygon", "Args":["38.33971392789639", "34.56548130065619", "L
oc-6"]}'
Point (38.33971, 34.56548) is INSIDE the polygon with ID: Loc-6
```

Ümit finally reaches the warehouse of the X Association in İskenderun and delivers the aid items to warehouse worker Emirhan Kızılırmak (Usr-3: Emirhan Kızılırmak, Loc-4: Antakya Warehouse of X Association).

```
ubuntu@aid-transfer-dev:~/aid-transfer/contracts/chaincode-external$
 peer chaincode invoke -C formal -n asset-transfer -c '{"function":"
TransferAid", "Args":["Aid-2", "Usr-2", "Usr-3","Loc-3","Loc-4"]}' |
 jq
2024-09-04 01:40:30.997 +03 0001 INFO [chaincodeCmd] InitCmdFactory
-> Retrieved channel (formal) orderer endpoint: orderer-api.127-0-0-
1.nip.io:8080
2024-09-04 01:40:31.031 +03 0002 INFO [chaincodeCmd] chaincodeInvoke
OrQuery -> Chaincode invoke successful. result: status:200
```

Additionally, Ümit checks whether he has brought the aid items to the correct location.

```
ubuntu@aid-transfer-dev:~/aid-transfer/contracts/chaincode-external$
 peer chaincode query -C formal -n asset-transfer -c '{"function":"P
ointInPolygon", "Args":["36.34662751700064", "36.20951332395217", "L
oc-4"]}'
Point (36.34663, 36.20951) is INSIDE the polygon with ID: Loc-4
```

After the planning within the association, Emirhan controls the distributors working in that region (Wor-2: Hatay - Antakya) to distribute the aid items.

```
ubuntu@aid-transfer-dev:~/aid-transfer/contracts/chaincode-external$
 peer chaincode query -C formal -n asset-transfer -c '{"function":"W
orkingAreaCheck", "Args":["Wor-2"]}' | jq
[
  {
    "id": "Usr-4",
    "actorType": "Distributor",
    "organization": "X Association",
    "department": "Assistance Board",
    "title": "Volunteer",
    "name": "Emre Özdilek",
    "contactNumber": "90 5317136317",
    "email": "eozdilek@xassociation.org",
    "workplaceID": "Wor-2",
    "remarks": ""
  }
]
```

Emirhan delivers the aid items to Emre Özdilek, who is the appropriate distributor (Usr-4: Emre Özdilek).

```
ubuntu@aid-transfer-dev:~/aid-transfer/contracts/chaincode-external$
 peer chaincode invoke -C formal -n asset-transfer -c '{"function":"
TransferAid", "Args":["Aid-2", "Usr-3", "Usr-4","Loc-4","Loc-4"]}' |
 jq
2024-09-04 01:42:49.363 +03 0001 INFO [chaincodeCmd] InitCmdFactory
-> Retrieved channel (formal) orderer endpoint: orderer-api.127-0-0-
1.nip.io:8080
2024-09-04 01:42:49.398 +03 0002 INFO [chaincodeCmd] chaincodeInvoke
OrQuery -> Chaincode invoke successful. result: status:200
```

Emre goes to Büyükdalyanlı Container City for distribution and when he gets there, he reports his location to the blockchain network.

```
ubuntu@aid-transfer-dev:~/aid-transfer/contracts/chaincode-external$
 peer chaincode query -C formal -n asset-transfer -c '{"function":"P
ointInPolygon", "Args":["36.2981936018108", "36.19603997119771", "Lo
c-2"]}'
Point (36.29819, 36.19604) is INSIDE the polygon with ID: Loc-2
```

After reporting his location, Emre delivers the aid items to Kader Samet, the camp manager (site authority) of the Disaster Coordination Center (Usr-5: Kader Samet, Loc-2: Büyükdalyan Temporary Accommodation Center - Hatay).

```
ubuntu@aid-transfer-dev:~/aid-transfer/contracts/chaincode-external$
 peer chaincode invoke -C formal -n asset-transfer -c '{"function":"
TransferAid", "Args":["Aid-2", "Usr-4", "Usr-5","Loc-4","Loc-2"]}' |
 jq
2024-09-04 01:44:13.567 +03 0001 INFO [chaincodeCmd] InitCmdFactory
-> Retrieved channel (formal) orderer endpoint: orderer-api.127-0-0-
1.nip.io:8080
2024-09-04 01:44:13.631 +03 0002 INFO [chaincodeCmd] chaincodeInvoke
OrQuery -> Chaincode invoke successful. result: status:200
```

Donor Ayşe checks where the aid items are. In this query, the "newLocationID" is Loc-2, meaning the diapers have reached the final destination, Büyükdalyan Temporary Accommodation Center - Hatay.

```
ubuntu@aid-transfer-dev:~/aid-transfer/contracts/chaincode-external$
 peer chaincode query -C formal -n asset-transfer -c '{"function":"W
hereIsMyDonation", "Args":["Trn-Aid-2"]}' | jq
{
  "id": "Trn-Aid-2",
  "aidID": "Aid-2",
  "previousActorID": "Usr-4",
  "newActorID": "Usr-5",
  "previousLocationID": "Loc-4",
  "newLocationID": "Loc-2",
  "transferDate": "0001-01-01T00:00:00Z",
  "remarks": ""
}
```

4.6. Discussion

Within the scope of this thesis, proof of location of aid delivery is particularly focused on. In this context, this location proof has been applied in various steps, but it is possible to make these applications even more advanced. For example, in the “In-kind aid ID generation” step, in addition to the proof provided by declaration and photo, the donor can read the sensor with a mobile device when they come to the organization branch and prove from another source that the aid was brought to the branch [43]. Since the buildings of the organizations are indoor areas, the location proof concept can also be supported with UWB (Ultra-wideband) or BLE (Bluetooth Low Energy), which are ideal methods for indoor location estimations [78,79]. Also, in the “Transporting the aid items to the target area” and “Local dispatching of aid to its destination” steps, in addition to the proof method made by point in polygon, photography, and declaration, various sensors can be deployed at stations or delivery points and readings can be made [43]. As in the previous example, the proof of location concept can be further strengthened for indoor areas by using UWB sensor technology or BLE [78,79].

It is possible to develop further details of "Organization building" and "Location", which are currently the data structures of the study. Fields containing data such as the number of floors of the building of the organization, net square meters, heating type, presence of a freight elevator, and the presence of any parking area for vehicles to carry the load can be added to the "Organization building" data structure. With the analysis obtained from this data, aid organizations can make more comprehensive plans.

Fields containing data such as the number of people living in that area, the type of shelter (container city, tent city, refugee camp, irregular settlement not affiliated with a formal organization, home, etc.), the number of shelters, and the needs of this area can also be added the "Location" structure. With the analyses obtained from these data, where and what the people in need can be addressed more comprehensively.

Additionally, adding a data structure for formal site areas, where site authorities can report needs, may also provide extremely useful results. Such a data structure means obtaining a very comprehensive needs assessment, as it allows organizations to learn first-hand needs.

5. CONCLUSION

This study focuses on the potential of using location-based smart contracts and blockchain technology to improve the transparency and traceability of in-kind aid delivery in disaster and migration monitoring scenarios. A concept of in-kind aid delivery is developed by analyzing the organizations, networks, and actors. A process is developed and the activities of each actor are determined by their information need and restrictions. The transactions of each in-kind aid item are identified with a focus on location-based transaction logic.

Hyperledger framework and Go programming language are utilized in the implementation of a proof of concept smart contract to support the concept development. The network design and authorization checks for each role are identified and data models are designed to support these checks. The Go language implementation of data models and proof of concept pseudo codes of location-based smart contract checks are also listed with some discussion in implementation detail. It has been concluded that with the development of supporting client applications such a process can be implemented with various checks based on location injected into the database to make the system traceable, transparent, and also fault tolerant.

Although smart contract based location checking may satisfy the tracking and checking of in-kind aid delivery to its intended target, the proof of location is a still major concern. Technologies such as near field communication, ultra wide band, indoor positioning, and GNSS integrity check services can all be considered for further development.

REFERENCES

[1] CAF (Charities Aid Foundation), "World Giving Index 2023," 2023.

[2] CAF (Charities Aid Foundation), "World Giving Index 2021," 2021.

[3] U. Gentilini, "Why does in-kind assistance persist when evidence favors cash transfers?," Brookings Institution, [Online]. Available: https://www.brookings.edu/articles/why-does-in-kind-assistance-persist-when-evidence-favors-cash-transfers/. [Accessed 8 March 2024].

[4] S. E. Aytaç and A. Çarkoğlu, "Individual Giving and Philanthropy in Turkey 2021," 2022.

[5] "Hyperledger Fabric," [Online]. Available: https://github.com/hyperledger/fabric. [Accessed 13 April 2024].

[6] IBM, "What is Hyperledger Fabric?," [Online]. Available: https://www.ibm.com/topics/hyperledger. [Accessed 13 April 2024].

[7] UN OCHA, "Global Humanitarian Overview 2024," 2023.

[8] LOSEV (Foundation for Children with Leukemia), "Ayni Bağış Açıklama...," [Online]. Available: https://www.losev.org.tr/bagis/Ayni_Bagis_Aciklama.htm. [Accessed 8 March 2024].

[9] TEMA (The Turkish Foundation for Combating Soil Erosion, for Reforestation and the Protection of Natural Habitats), "Ayni Bağış," [Online]. Available: https://www.tema.org.tr/bagis-ve-destek/ayni-bagis. [Accessed 8 March 2024].

[10] Darüşşafaka Society, "Donations in kind," [Online]. Available: https://www.darussafaka.org/en/your-donations/donation-types/material-donations. [Accessed 8 March 2024].

[11] KASDER (Turkish Muscle Diseases Association), "Ayni Yardımlar," [Online]. Available: https://kasder.org.tr/hizmetlerimiz/hizmetlerimizin-listesi/ayni-yardimlar/. [Accessed 8 March 2024].

[12] Ahbap Association, "Tüzük," [Online]. Available: https://ahbap.org/tuzuk. [Accessed 8 March 2024].

[13] THK (Turkish Aeronautical Association), "AYNİ BAĞIŞ," [Online]. Available: https://www.thk.org.tr/ayni_bagis. [Accessed 8 March 2024].

[14] Yeşilay (Turkish Green Crescent Society) , "Ayni Bağışlar," [Online]. Available: https://www.yesilay.org.tr/tr/ayni-bagis. [Accessed 8 March 2024].

[15] Türk Kızılay (Turkish Red Crescent Society) , "Diğer Bağış Yöntemleri," [Online]. Available: https://www.kizilay.org.tr/bagis-yontemleri/diger-bagis-yontemleri. [Accessed 8 March 2024].

[16] İHH (Humanitarian Relief Foundation), "Ayni bağış kabul ediyor musunuz?," [Online]. Available: https://ihh.org.tr/sss/ayni-bagis-kabul-ediyor-musunuz. [Accessed 8 March 2024].

[17] B. Singhal, G. Dhameja and P. S. Panda, Beginning Blockchain: A Beginner's guide to building Blockchain solutions, vol. 1, Apress, 2018.

[18] S. Bearman, "Bitcoin's creator may be worth $6 billion — but people still don't know who it is," CNBC, [Online]. Available: https://www.cnbc.com/2017/10/27/bitcoins-origin-story-remains-shrouded-in-mystery-heres-why-it-matters.html. [Accessed 18 March 2024].

[19] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://www.metzdowd.com/pipermail/cryptography/2008-October/014810.html. [Accessed 19 Mar 2024].

[20] S. S. Sarmah, "Understanding Blockchain Technology," *Computer Science and Engineering,* vol. 8, pp. 23-29, 2018.

[21] M. A. Al-Shabi, "A Survey on Symmetric and Asymmetric Cryptography Algorithms in information Security," *International Journal of Scientific and Research Publications (IJSRP),* vol. 9, 2019.

[22] M. A. Panhwar, S. A. Khuhro, G. Panhwar and K. A. Memon, "SACA: A Study of Symmetric and Asymmetric Cryptographic Algorithms," *International Journal of Computer Science and Network Security (IJCSNS),* vol. 19, pp. 48-55, 2019.

[23] S. Rani and H. Kaur, "Technical Review on Symmetric and Asymmetric Cryptography Algorithms," *International Journal of Advanced Research in Computer Science,* pp. 182-186, 2017.

[24] A. Usta and S. Doğantekin, Blockchain 101, vol. 2, BKM (Interbank Card Center), 2018.

[25] C. Say, "5 Soruda Blok Zinciri," BKM (Interbank Card Center), 2019.

[26] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," *Journal of Cryptology,* vol. 3, no. 2, pp. 99-111, 1991.

[27] D. Tapscott and A. Tapscott, Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World, Penguin, 2016.

[28] N. Gupta, "Symmetric vs. Asymmetric Encryption – What are differences?," [Online]. Available: https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences. [Accessed 10 April 2024].

[29] Ethereum Foundation, "Introduction to smart contracts," [Online]. Available: https://ethereum.org/smart-contracts/. [Accessed 19 March 2024].

[30] N. Szabo, "Smart Contracts," 1994. [Online]. Available: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html. [Accessed 28 07 2024].

[31] N. Szabo, "Smart contracts: building blocks for digital markets," 1996. [Online]. Available: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html. [Accessed 28 07 2024].

[32] L. Luu, D.-H. Chu, H. Olickel, P. Saxena and A. Hobor, "Making Smart Contracts Smarter," in *CCS '16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna, 2016.

[33] The Solidity Authors, "Introduction to Smart Contracts," [Online]. Available: https://docs.soliditylang.org/en/latest/introduction-to-smart-contracts.html. [Accessed 11 April 2024].

[34] M. Di Pierro, "What is the blockchain?," *Computing in Science & Engineering,* vol. 19, pp. 92-95, 2017.

[35] M. Kandpal, V. Goswami, R. Priyadarshini and R. K. Barik, "Towards Data Storage, Scalability, and Availability in Blockchain Systems: A Bibliometric Analysis," *Data,* vol. 8, no. 10, p. 148, 2023.

[36] C. Dwork and M. Naor, "Pricing via Processing or Combatting Junk Mail," in *Advances in Cryptology — CRYPTO' 92*, Berlin, 1993.

[37] M. Jakobsson and A. Juels, "Proofs of Work and Bread Pudding Protocols," in *Secure Information Networks: Communications and Multimedia Security IFIP TC6/TC11 Joint Working Conference on Communications and Multimedia Security (CMS'99)*, Boston, 1999.

[38] S. Chamanara, S. A. Ghaffarizadeh and K. Madani, "The Environmental Footprint of Bitcoin Mining Across the Globe: Call for Urgent Action," *Earth's Future,* vol. 11, no. 10, 2023.

[39] P. Davies, "Europe rejects proposal limiting PoW cryptos such as Bitcoin but sets draft rules for sustainability," Euronews, [Online]. Available: https://www.euronews.com/next/2022/03/14/europe-to-vote-on-limiting-pow-crypto-mining-used-by-bitcoin-and-ethereum. [Accessed 10 April 2024].

[40] Ethereum Foundation, "Staking," [Online]. Available: https://ethereum.org/en/staking/. [Accessed 7 April 2024].

[41] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao and M. Imran, "A Scalable Multi-layer PBFT Consensus for Blockchain," *IEEE Transactions on Parallel and Distributed Systems,* pp. 1146-1160, 2020.

[42] G. Brambilla, M. Amoretti and F. Zanichelli, "Using Block Chain for Peer-to-Peer Proof-of-Location," *ArXiv,* 2016.

[43] R. Khan, S. Zawoad, M. M. Haque and R. Hasan, "Who, When, and Where?’ Location Proof Assertion for Mobile Devices," in *Data and Applications Security and Privacy XXVIII*, Vienna, 2014.

[44] P. Gilbert, L. P. Cox, J. Jung and D. Wetherall, "Toward trustworthy mobile sensing," in *HotMobile '10: Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*, Annapolis, Maryland, 2010.

[45] A. Trouw, M. Levin and S. Scheper, "The XY Oracle Network: The Proof-of-Origin Based Cryptographic Location Network," 2018. [Online]. Available: https://docs.xyo.network/XYO-White-Paper.pdf. [Accessed 11 April 2024].

[46] "Proof of Location," [Online]. Available: https://tokens-economy.gitbook.io/consensus/chain-based-proof-of-capacity-space/dynamic-proof-of-location. [Accessed 11 April 2024].

[47] S. S. Sahoo and V. K. Chaurasiya, "Proof of location based delivery system using multi-party virtual state channel: a blockchain model," *The Journal of Supercomputing,* vol. 80, no. 1, pp. 703-733, 2024.

[48] H. R. Hasan and K. Salah, "Blockchain-Based Proof of Delivery of Physical Assets With Single and Multiple Transporters," *IEEE Access,* vol. 6, pp. 46781-46793, 2018.

[49] Hyperledger Foundation, "An Introduction to Hyperledger," [Online]. Available: https://8112310.fs1.hubspotusercontent-na1.net/hubfs/8112310/Hyperledger/Offers/HL_Whitepaper_IntroductiontoHyperledger.pdf. [Accessed 13 April 2024].

[50] S. Namasudra, G. Deka, P. Johri, M. Hosseinpour and A. Gandomi, "The Revolution of Blockchain: State-of-the-Art and Research Challenges," *Archives of Computational Methods in Engineering,* vol. 28, 2020.

[51] Hyperledger Foundation, "Introduction," [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/whatis.html. [Accessed 14 April 2024].

[52] Hyperledger Foundation, "Welcome Hyperledger Fabric 2.0: Enterprise DLT for Production," [Online]. Available:

https://www.hyperledger.org/blog/2020/01/30/welcome-hyperledger-fabric-2-0-enterprise-dlt-for-production. [Accessed 14 April 2024].

[53] J. Polge, J. Robert and Y. Le Traon, "Permissioned blockchain frameworks in the industry: A comparison," *ICT Express,* vol. 7, pp. 229-233, 2020.

[54] S. Cocco and G. Singh, "Top 6 technical advantages of Hyperledger Fabric for blockchain networks," IBM, [Online]. Available: https://developer.ibm.com/articles/top-technical-advantages-of-hyperledger-fabric-for-blockchain-networks/. [Accessed 13 April 2024].

[55] Hyperledger Foundation, "How Fabric networks are structured," [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/network/network.html. [Accessed 12 May 2024].

[56] Hyperledger Foundation, "Membership Service Provider (MSP)," [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/membership/membership.html. [Accessed 12 May 2024].

[57] Hyperledger Foundation, "Peers," [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/peers/peers.html. [Accessed 13 May 2024].

[58] Hyperledger Foundation, "The Ordering Service," [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/orderer/ordering_service.html. [Accessed 13 May 2024].

[59] UN WFP, "Building Blocks Frequently Asked Questions (FAQs)," [Online]. Available: https://docs.wfp.org/api/documents/WFP-0000146542/download/. [Accessed 15 March 2024].

[60] UN WFP, "Report, Building Blocks Ukraine Unintended Assistance Overlap Prevention," 2022.

[61] UN WFP, "Building Blocks - Blockchain network for humanitarian assistance - Graduated Project," [Online]. Available: https://innovation.wfp.org/project/building-blocks. [Accessed 14 Mar 2024].

[62] UN WFP, "Building Blocks," [Online]. Available: https://www.wfp.org/building-blocks. [Accessed 15 Mar 2024].

[63] V. Poonia, M. K. Goyal, B. Gupta, A. K. Gupta, S. Jha and J. Das, "Drought occurrence in Different River Basins of India and blockchain technology based framework for disaster management," *Journal of Cleaner Production,* vol. 312, 2021.

[64] T. G. Türkdoğan, "An Evaluation on The Applicability of Block Chain Technology in Disaster and Risk Management," *Atlas Journal,* vol. 10, no. 53, pp. 23-40, 2024.

[65] H. Baharmand and T. Comes, "Leveraging Partnerships with Logistics Service Providers in Humanitarian Supply Chains by Blockchain-based Smart Contracts," *IFAC-PapersOnLine,* vol. 52, no. 13, 2019.

[66] M. Demir, O. Turetken, A. Ferworn and M. Kargar, "A Blockchain-Based System for Aid Delivery: Concept Development, Data Modeling, and Validation," *Journal of Database Management,* vol. 34, 2023.

[67] M. Demir, O. Turetken and A. Ferworn, "Blockchain-Based Transparent Disaster Relief Delivery Assurance," in *IEEE International Systems Conference (SysCon)*, Virtual, 2020.

[68] M. Bal, "Securing property rights in India through distributed ledger technology," *Observer Research Foundation,* 2017.

[69] M. Velpuri, S. Bhamidipati, M. A. Sharma, C. Maringanti, Y. H. Chin, A. Pidugu and J. Velpuri, "Fintech for geo-spatial transformation and Real estate management," in *FIG Congress*, İstanbul, 2018.

[70] R. Cedeno, Z. Pengxiang, A. Mansourian and M. Brovelli, "Geospatial Blockchain: review of decentralized geospatial data sharing systems," *AGILE: GIScience Series,* vol. 3, pp. 1-6, 2022.

[71] A. F. Mendi, Ö. Demir, K. K. Sakaklı and A. Çabuk, "A new approach to land registry system in Turkey: blockchain-based system proposal," *Photogrammetric Engineering & Remote Sensing,* vol. 86, no. 11, pp. 701-709, 2020.

[72] Q. Zou, W. Yu and Z. Bao, "A Blockchain Solution for Remote Sensing Data Management Model," *Applied Sciences,* vol. 13, 2023.

[73] M. Pincheira, E. Donini, R. Giaffreda and M. Vecchio, "A Blockchain-Based Approach To Enable Remote Sensing Trusted Data," in *2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS)*, Santiago, 2020.

[74] IOM - UN Migration, "DISPLACEMENT TRACKING MATRIX│DTM," [Online]. Available: https://www.iom.int/sites/g/files/tmzbdl486/files/our_work/DOE/humanitarian_emergencies/IOM-DTM-Infosheet.pdf. [Accessed 16 Mar 2024].

[75] https://www.basarsoft.com.tr/tematik-harita-nedir/, "Tematik Harita Nedir?," [Online]. Available: https://www.basarsoft.com.tr/tematik-harita-nedir/. [Accessed 16 Mar 2024].

[76] IOM - UN Migration, "Solution and Mobility Index," [Online]. Available: https://dtm.iom.int/solutions. [Accessed 29 Mar 2024].

[77] IOM - UN Migration, "DTM Türkiye — Migrant Presence Monitoring - Annual Migrant Presence Monitoring (Jan-Dec 2022)," 2023.

[78] T. Arsan, "Improvement of indoor positioning accuracy of ultra-wide band sensors by," *Pamukkale University Journal of Engineering Sciences,* vol. 24, no. 5, pp. 921-928, 2018.

[79] O. Eroğlu and M. Doğan, "Konum Tabanlı Hizmetler, Bluetooth Tabanlı İç Mekan Konumlandırma Hizmeti, Antalya İli Örneği," in *TMMOB 6. Coğrafi Bilgi Sistemleri Kongresi*, Ankara, 2019.