A Blockchain-Based System for Aid Delivery: Concept Development, Data Modeling, and Validation

Mehmet Demir, Toronto Metropolitan University, Canada
Ozgur Turetken, Toronto Metropolitan University, Canada
Alexander Ferworn, Toronto Metropolitan University, Canada
Mehdi Kargar, Toronto Metropolitan University, Canada*

ABSTRACT
Climate-related catastrophes leave people in dire need of aid. A major obstacle in providing help to people is the lack of trust in the aid process. Charity organizations want to ensure that funds and materials reach the intended destinations. Blockchain technology injects trust into business transactions through impeccable record keeping and can alleviate the trust problems in aid delivery. Another major problem in disaster recovery is broken infrastructure (e.g., broken bridges and unavailable roads). Unmanned aerial vehicles (UAV), generally referred to as drones, can address this access problem. In this paper, the authors design a system that uses drone technology for delivery of aid and blockchain technology for the assurance of such delivery. This system records and shares data on the interaction of various participants involved in a disaster aid delivery scenario. The simulation studies validate the applicability of this proposed system showing high throughput and satisfactory performance are attainable with integration of blockchain in large-scale aid delivery.

KEYWORDS
Aid Delivery, Blockchain, Data Flow Management, Data Model for Delivery, Disaster Management, Drone Technology

1. INTRODUCTION
Climate-related catastrophes and wars leave people in dire need of aid. Disasters occur suddenly, cause severe disturbance, inflict pain, and disrupt the lives of a great many people. Disaster relief needs to be agile, quick, and in high volume while being provided through challenging physical conditions (e.g., after a severe flood or earthquake). One of the major drivers of these challenging
conditions is the disruption of the usual methods and channels of delivery. In the United States alone, Hurricane Catrina (Scowcroft, 2020), Hurricane Harvey (Hernández, Zezima, and Achenbach, 2017) and Hurricane Maria (Jenkins, 2017), among many others, have shown that even if aid is ready to be delivered, closed roads, broken infrastructure, and depleted human resources prevent immediate relief. Drone technology is promising in providing an alternative to the conventional truck-and-person delivery model. Instead of using roads, drones take advantage of the clear skies. This paradigm shift is proven effective in disaster conditions where the roads are blocked by debris and where there are no roads built due to poverty (Doole, Ellerbroek, and Hoekstra, 2020).

Even if the broken infrastructure challenges are circumvented by the use of drones, another problem relief delivery still suffers from is trust. Many otherwise charitable individuals and organizations may refrain from donating to disaster relief efforts or contributing aid materials, because they are doubtful that the aid will reach the intended parties. What causes such lack of faith is the chaos inherent in a disaster scenario, which renders the typical verification of transactions infeasible. By its nature, blockchain technology is a good candidate to address such trust issues. By providing a distributed platform where no authority can single-handedly decide to rewrite the recorded information, blockchain implementations provide a single source of truth for each stakeholder (Queiroz and Samuel, 2018; Yong, et al., 2020). This is the main advantage of a blockchain based system design over those that use more traditional data storage models such as RDMS and NoSQL.

In this paper, we report on the design of a system that delivers aid supplies through drone technology and records the delivery events into a blockchain in a disastrous hurricane scenario. In this scenario, the flood following the hurricane blocks roads with debris and muddy flood waters. People are trapped in or on their houses in desperate need of emergency supplies such as drinkable water or first aid materials. In such a scenario, both the physical access and trust problems are present, and we address these problems by a unique combination of drone and blockchain technologies. As such, we aim to address the following research questions:

**RQ1:** How can a framework be developed to guide governments and relief organizations in aid delivery during a disaster when roads are blocked?

**RQ2:** How can blockchain technology manage the data flow of drone-based delivery processes during aid delivery for disaster relief?

The rest of this paper is organized as follows. A discussion on related work and current research gap are presented in the next section. Details of the problem studied, and our novel solution are presented as well as results from testing the proposed solution through simulations are presented in Section 3. Section 4 discusses the theoretical and practical contributions of the paper, and Section 5 concludes the paper with implications of the work reported herein and directions for future research.

### 2. REVIEW OF EXISTING WORK

Although blockchain technology gained widespread popularity mainly as a platform for virtual currency implementations, it can also be used in a variety of other applications to build information systems (Subramanian and Liu, 2021). For example, Mashatan, et al., (2021) study the problem of double-ending fraud in real estate transactions and propose a blockchain-based solution to resolve it. Thakur and colleagues discuss the application of blockchain for land record management in India (Thakur, Doja, Dwivedi, Ahmad, and Khadanga, 2020). Li, et al., (2017) present a secure energy trading system using blockchain that can deal with non-truthful energy markets. These recent examples from the information systems literature verify the versatility of blockchain technology in a variety of applications. As a delivery problem, what we focus on in this research is in the general category of supply chain management, for which there is a substantial sample of blockchain applications (Marr, 2018).
2.1 Blockchain and Supply Chain Management

Most industrial supply chain management systems currently in use are designed as centralized systems. There is a heavy burden on the central authority to collect information, recognize events, and take actions such as billing and payments. Especially the logistics component of supply chain management has several use cases where blockchain technology can change the business. Logistics processes involve several interacting parties. The trading partners interact with each other and experience the difficulties and challenges related to the level of information shared. This information asymmetry creates inefficiencies for the business environment. Because of the continuous conflict of interest situations in trade, blockchain is a natural solution (Nakasumi, 2017). Business events in the supply chain can be recorded and stored in a blockchain utilizing all the features of a distributed ledger. Further, when certain events happen, smart contracts can be executed automatically to manage payments. Companies at distributed locations can conduct business where all events and actions are recorded immutably without the need for traditional trust or central management.

The literature indicates the need for a reliable and permanent tracking system for resources such as vehicles deployed for distribution. As long as such a reliable system is available, technologies such as GPS can be integrated as a source of input information into these systems (Tian, 2017). There is a great desire for analytics aspects of data, and the only way to collect reliable data for this purpose is with a trustable technology such as blockchain (Kamble, Gunasekaran, and Arha, 2018).

Due to its conceptual appeal as discussed above, the application of blockchain in supply chain management and delivery has gained increasing research attention over the last decade. Raj and Sowmiya (2021) present a detailed discussion on the use of blockchain technology in the field of supply chain management. To analyze the impact of blockchain technology for supply chain management and the interaction of different actors, Tonnissen and Teuteberg (2020) present an explanatory model drawn from multiple case studies. According to this study, traditional intermediaries keep part of their previous roles as a connector between producer and consumer, and therefore will exist in blockchain-based supply chain management. However, some roles of intermediaries can be performed by existing producers in a blockchain service provider. The blockchain service provider enables its customers to integrate existing ICT infrastructure with blockchain technology. It also offers other services to its customers such as customization of the platform, consulting, and implementation for specific customers’ needs. Kleinknecht (2021) focuses on the use of blockchain in Sustainable Supply-Chain Governance. The author explores the role of blockchain in environmental regulations and standards in corporate sustainable supply-chain practices.

Agriculture supply chain management is critical for transporting the basic needs of human beings. In a number of regions across the globe, the agriculture supply chain is problematic and not well managed. Hence, farmers do not know the real condition of their goods upon delivery. Furthermore, farmers do not have much negotiation power in terms of the delivery costs. Blockchain technology could change this as it has the potential to offer security and immutable storage of data. Sudha et al., 2021 analyze different strategies proposed for deploying blockchain for the agriculture supply chain management and discuss the advantages and disadvantages of each strategy.

Yong, et al., (2020) develop a blockchain-based system for safe vaccine delivery. This system provides support for vaccine traceability and smart contract functions. It is also able to deal with vaccine record fraud and vaccine expiration. A framework for controlling energy supply and delivery with blockchain is discussed in Andoni, Robu, and Flynn (2017). The benefits of a blockchain system in offering decentralization and traceability is taken into account in this system. Ahmed and Broek (2017) introduce a system for secure food delivery with blockchain. According to the authors, food security could significantly benefit from the transparency, relatively low transaction costs, and instantaneous applications of the blockchain technology. Behnke and Janssen (2020) propose a blockchain-based framework for food supply chain management with traceability and study several boundary conditions such as standardization of traceability processes and interfaces, independent governance, and having a joint platform. Herbaut and Negru (2017) present the application of blockchain smart contracts in a video delivery ecosystem.
2.2 Adoption of Blockchain-Based Solutions

Schuetz and Venkatesh (2020) propose that blockchain technologies have the potential to overcome many challenges such as accessibility that organizations and their stakeholders face but argue that an understanding of technology adoption is of paramount importance for blockchain to be considered as a viable system component. A number of recent studies investigated the adoption of blockchain based solutions especially in supply chain and delivery.

Kamble, Gunasekaran, and Sharma (2020) identify the relationships between the enablers of blockchain adoption in agriculture supply chains. The authors find that traceability is the most significant reason for blockchain implementation. Other important enablers include auditability, immutability, and provenance. Queiroz and Samuel (2018) study the individual blockchain adoption in supply chain management in India and the USA. The authors design a model based on a modified version of the classical unified theory of acceptance and use of technology. The results of this work indicate the existence of distinct adoption behaviors among India-based and USA-based experts. Furthermore, their results show that blockchain adoption by experts in the fields of logistics and supply chain management is still in its early stages, and deeper investigation is required for blockchain based systems to be mainstream in the industry.

The work of Wong, Leong, Hew, Tan, and Ooi (2020) studies the effects of different factors (such as market dynamics, competitive pressure, and regulatory support) on blockchain adoption for supply chain management and operation among small and medium enterprises (SME) in Malaysia. A comprehensive framework is proposed that covers the organizational dimensions of higher management cost, support of regulation, environmental aspects of market dynamics, and technological aspects. This research concludes that SMEs usually do not have sufficient support for technological investment but have the same requirements to streamline different aspects of their business. Therefore, blockchain provides a valuable option for continues progress and business growth to SMEs because it has the potential to offer immutability, transparency, and security.

Biswas and Gupta (2019) design a framework for investigating barriers to the adoption and implementation of blockchains across different industries. These barriers are identified with the help of extant literature and experts’ opinions. They are then classified into ten broad categories. Authors then detect the causal relationships among those barriers and rank them based on their degree of prominence and relationships. The result of this finding suggests that challenges in market-based risks and scalability are the most important barriers. On the other hand, high sustainability costs and poor economic behavior are factors that affect this the least. A number of sensitivity analysis is performed to test the effectiveness of the proposed framework. The result of this work could be useful for decision makers that want to detect and eliminate barriers in the efficient implementation of blockchain technologies.

In the context of fashion industry, Caldarelli et al. (2021) examine and overcome the barriers to the widespread adoption of blockchain technology. The authors introduce a novel concept of sustainability in the fashion supply chain. Data collected from interviews with different groups of stakeholders in the fashion industry in Italy suggests that a blockchain solution could potentially contribute to sustainable supply chains. However, a good understanding of the core technology and communication with different clients and customers are necessary to have a seamless solution.

Hofmann et al. (2018) study the application of blockchain in supply chain finance, and specifically focus on in approved payables financing. After identifying the main barriers in delivering financing solutions, the authors propose a possible blockchain-based supply chain framework that offers benefits such as improved business processes and lower costs of financing to all parties involved in a supply chain finance transaction.

2.3 Research Gaps

As long established by the information systems literature, information technology success depends on both the successful implementation of technology and the adoption of that implementation by users
(Goodhue and Thompson, 1995). The research reviewed in the previous sections addresses the latter (i.e., adoption) problem. Meanwhile, few blockchain applications reported in the literature go beyond the description of a system design, therefore the first condition of system success, i.e., successful implementation of technology, is not always established. In order for supply chain management to adopt blockchain, supply chain managers need to understand benefits, compatibility with current practices, complexity of the usage, ease of testing, and the provision of visible results (Dobrovnik, Herold, Fürst, and Kummer, 2018). Our use case and simulation experiments presented in this paper provide this deep understanding. As such, this paper not only describes a system design for the important problem of aid delivery, but also establishes the validity and feasibility of that design through simulation experiments. We also share the code for our proof-of-concept system, share the metrics, and keep the project active for further research.

There are several commercial projects addressing delivery assurance using blockchain technology. IBM Developer community (Mohan, 2017) has one of the most descriptive blockchain architectures related to disaster recovery. This blockchain solution is fundamentally modeled as a use case on IBM platforms. It is a general architecture that describes a complicated solution for a focused problem. It is neither a supply chain nor a blockchain solution framework. Without implementation, the use case seems to be unrealistic since it includes medical records and videos recorded by users during a crisis. It also does not record the internet of things (IoT) device data.

Hybrid peer-to-peer physical distribution (HP3D) is a ledger architecture for supply chain distribution visibility (Wu, et al., 2017). This ledger architecture models the information flow between supplier, distribution center, customer, and carrier. The specific solution promoted in this research does not support broad participation and complete transparency.

The commercial system by Modum\(^1\) stores sensor data for pharmacy products. Even though the blockchain implementation is added to the system, it is backed with traditional systems due to the sensitive nature of data. The dependency on traditional systems makes it a centralized solution that does not benefit from a decentralized architecture.

The TradeLens\(^2\) platform, another blockchain platform by IBM, focuses on shipping. TradeLens manages conventional paper shipping processes including all intermediary steps at factories, land transports, ports, ships, ports, customs, and warehouses through blockchain events. It is a solution implemented exclusively for the shipping industry and does not focus on the last mile. The solution is implemented, and dependent, on IBM platforms. It does not consider crowdsourcing or autonomous vehicles. TradeLens does not have the ability to be customized for disaster recovery. The main solution scenarios (such as inspection and transporting dangerous goods) are all shipping related.

Blockchain technology is fundamental for distributed trust. However, current commercial implementations of blockchain technology have yet to create entirely distributed systems. Furthermore, a significant difference between our approach and projects that use specific blockchains such as Ethereum is that such systems inherit all the limitations of those blockchains. In order to reach the flexibility of the data model and agility, we avoid such vendor dependency. The products that IBM provides are well integrated into the rest of the IBM Cloud. However, this single platform mentality and integration reduces the flexibility of these solutions. We provide a solution with the ability to be implemented on any blockchain platform.

Studies such as Hasan and Salah (2018) that focus on the decentralized marketplaces where sellers and buyers meet, and payments are conducted in Ethereum cryptocurrency tokens do not have a flexible model to store data on the blockchain due to the Ethereum blockchain design. Instead, they rely on additional systems such as Inter Planetary File System (IPFS) to store the data while the blockchain only stores the information hash. We believe decentralization opens the path to crowdsourcing related innovations. Current crowdsourcing projects still require a trusted entity. For example, the rideshare system Uber provides trust to users of its service. We are not intentionally removing these trust organizations from the solutions. However, our solution for delivery makes sure that the system is flexible for including an unlimited number of partners without central management.
Another main gap in the literature is the lack of IoT blockchain integration. Frameworks that model the delivery business with IoT monitoring events are very rare. Where they exist, they are limited to one type of attribute such as GPS or Temperature. Designing a suitable data model that provides required functionality is an important aspect in system design (Kazi and Kazi, 2019; Chan, Tan, and Teo, 2014). Our model does not limit the type of events and keeps the data model open for more monitoring. This is vital for sensitive deliveries that need the trust provided by blockchain.

Another critical point that is rare in the literature is the autonomous vehicles and other autonomous agents. The modeling of these vehicles is somewhat unique since they lack the human element. One exception is the GaRuDa framework by Gupta et al. (2021), which is a blockchain-based drone delivery scheme. Like the solution we propose, the Gupta et al. system integrates drones and blockchain using 5G-enabled tactile internet that support low-latency for delivery of medical equipment and supplies. The stated purpose of the reported solution is secure delivery of supplies in the presence of non-trusted communication channels, which is accomplished through the use of 5G, rather than blockchain, technology. In the absence of a detailed elaboration of trust issues between participants in the supply chain, the value of blockchain in this framework is not entirely clear.

Prior research e.g., Biswas, B., & Gupta, G. (2019) indicates that trust is not considered as a driver or barrier for blockchain adoption, which suggests that trust is taken for granted as a feature of the technology. What is still unknown and still debated though is the scalability of the technology. Blockchains are notoriously slow and scalability is a challenge. Hence limited scalability has been identified as a drawback to blockchain technology for its productive use for growing volumes of transactions (Biswas & Gupta, 2019; Caldarelli et al., 2021; Hofmann et al., 2018; Lohmer & Lasch, 2020; Tribis et al., 2018; Vafadarnikjoo et al., 2021). Our design aims to fill this gap by testing whether blockchain technology can scale to a problem like the one we identify in this paper even though it is generally accepted it improves trust in the underlying data through transparency and decentralization.

3. PROBLEM AND SOLUTION DETAILS

In this section, we present details of the problem and our proposed solution. Our research approach is naturally design science research as discussed by Venerable et al. (2017). We adopt the basics of the design science research process from Peffers, et al. (2006) as it fits naturally to the process we followed in our research and hence is a useful framework to structure of this section of the paper.

3.1 Problem Identification and Motivation

As discussed in the Introduction section of the paper, a major challenge to disaster relief is the disruption of the usual methods and channels of delivery. There are alternatives to standard infrastructure such as Unmanned Aerial Vehicle (UAV) technology that can be used in the aftermath of a disaster yet such alternatives still do not circumvent the issue that individuals and organizations may refrain from donating to disaster relief efforts or contributing aid materials, because they are doubtful that the aid will reach the intended parties. What causes such lack of faith is the chaos inherent in a disaster scenario, which renders the typical verification of transactions infeasible.

Immediately after a disaster’s destructive powers leave the impact area, disaster relief efforts start. Food, drinking water and first aid kits are some of the primary necessities for survivors. From this list, drinking water is the top item that victims need urgently. Water is the most significant single component of the human body, where 50%–60% of total body mass is water. It has a quick turnover of 2–3 liters. If the loss of water reaches 10–15% of the body mass, about 20–30% of the total body water, death is a likely outcome. This means that two days or more without water will have lethal consequences. Delivering fresh water is essential. This type of delivery happens using conventional means where applicable. Hurricane Maria has shown that when the delivery channels are obstructed with debris and floodwater, having a lot of water in the distribution centers does not bring any benefit to those in need. Months after Hurricane Maria bottled water sent to Puerto Rico was still sitting.
at the airports. Water is a heavy item to carry; it is not always logistically possible to provide high quantities to a high number of destinations.

3.2 Objective of a Solution
The objective of this work is to design a system for aid delivery in the immediate aftermath of a disaster. This framework should be reliable, trustworthy, and efficient. The aid (drinkable water) should be delivered even when roads are not accessible (e.g., due to flooding or fire). Tracking and verification of aid distribution should be part of the framework, so everyone affected gets a fair amount of aid, and no aid is stolen or wasted due to corruption or mismanagement.

3.3 Design and Development
3.3.1 Aid Items
There are more practical alternatives to providing the water itself during disasters such as a flood. The portable filtration kits or personal filtration devices are such alternatives. We have identified the aid items in Figure 1 as the most compatible with disaster delivery scenarios due to their weight and utility. Life-straw\textsuperscript{1} filters water removing bacteria, parasites and microplastics. It is durable and ultralight weighing only 57 grams. It has years-long shelf-life and can be used actively for months. Cleansip\textsuperscript{2} is a similar device but a lot lighter at 9.07 grams with a long shelf-life. Drinkable book\textsuperscript{5} is a booklet of water filtering papers that can be used to produce clean water from muddy and dirty water.

3.3.2 Crisis Centers
The source location and destination location are essential entities in any delivery scenario. Physical delivery in our context is the activity of transporting the deliverable from the source location to the destination location. Since we are focusing on the last mile of the delivery, our starting points are the crisis centers where drones pick up the aid materials and start the delivery. Every state/province and city has designated emergency gathering locations with a limited set of infrastructure support features such as power generators. Community centers, parks and large parking lots are examples of designated locations that are suitable to become distribution centers. Many government buildings except for fire stations can serve this purpose\textsuperscript{5}. Sports fields are also good candidates since they provide open areas with flat grounds.

Figure 1.
Example aid items: filtration devices
The second set of possible candidates for crisis centers is airports. Airports are designed to be air and land transportation hubs. They are almost always first to recover and go back to normal operations for aid to arrive and for people to evacuate. In the aftermath of hurricane Maria, in about 36 hours, San Juan airport was operational (Meyer, 2017). Airports are good candidates as bigger air vehicles bring in people, generators, and aid material while drones deliver these resources. The main disadvantage of airports is usually the restrictions posed on drones around airports. We can assume that the restrictions do not apply during disaster conditions, and drones can fly from relatively safe areas around the airport.

3.3.3 Drone Delivery

An important aspect of the solution we are proposing is the use of Unmanned Aerial Vehicle (UAV) technology. UAVs, also known as drones, carry out the delivery operations autonomously. While the operations are in progress, the drones communicate with the blockchain and issue transactions for the information that they are programmed to add to the blockchain. There are multiple categories of drones. For our problem, we are mainly interested in those that can deliver aid by taking frequent trips between the crisis center and the delivery destinations. Racing drones are the fastest drones in this category reaching to 260 km/hr (Segarra, 2017). There are several examples of delivery drones flying up to 100 km of distance with 100 km/hr speed (Corrigan, 2019). Although there are faster drones with longer range, we take the 100 km/hr speed and 100 km range as our base for the calculations.

3.3.4 Transaction Management via Blockchain Technology

By its nature, blockchain technology is a good candidate to address trust issues inherent in aid delivery. Through their decentralized platform where no authority can single-handedly change recorded information, blockchain implementations provide a single source of truth for each stakeholder (Queiroz and Samuel, 2018; Yong, et al., 2020). Therefore, our proposed solution records the delivery events into a blockchain. Below are the components of the blockchain-based design.

3.3.4.1 Participants

As part of an aid blockchain, Government, Aid Agencies, Auditors, Insurance Companies and Service Providers collaborate. For a permissioned blockchain, the number of members does not significantly contribute to quality. More members mean more information flows to the blockchain. Since we created this blockchain for the reliable collection of information, more information is beneficial and valuable. Participants of the network are depicted in Figure 2.

3.3.4.2 Data Model

Our data model is flexible, which gives us the ability to avoid privacy issues such that if the data is public, then it can be kept in the blockchain. If the data is large or private, it can be saved in Java Script Object Notation (JSON) based lightweight linked object standard (JSON-LD). This methodology also allows encryption that can be used in the data store or on the blockchain hosted data. Among all data hosting options, one option that we avoid is the central database that most applications seem to be based on. Instead of having a central database to speed up the system, we prefer to focus on the performance bottlenecks of the blockchain and provide a solution to those.

We use the record structures in the system to add information to our blockchain. Entities in the data model are represented as JSON objects in the smart contracts. When a client issues a transaction, all the information related to this transaction is sent to the blockchain node. When the blockchain receives the transaction, the smart contract engine of the blockchain identifies the transaction type, and the execution is forwarded to the corresponding smart contract code. The smart contract first creates JSON objects corresponding to the information in the transaction. Then smart contract entities are stored in the blockchain in their JSON representation.
Each entity we used in the aid blockchain is detailed in the following subsections. Entity representations are in Go language. Capital letter usage on some of the naming is part of the Go language scope rules. Each entity is represented with a “struct”, which is similar to a class definition in most object-oriented languages. Each field is defined by its name, its type and other annotation tags. Each line describes one field. JSON tags are also provided for each entity for the smart contract libraries to operate on the fields as needed.

3.3.4.2.1 Service Request

The service request entity represents a service that has been requested by victims or any other organization that is aware of the need for aid delivery. If the victims have a network connection, it is possible for them to create their own service requests. A more plausible case of service request creation would be either aid agencies or disaster management agencies to create these records. Each record includes the type of service that is requested, coordinates of the destination, status, and timestamps.

3.3.4.2.2 Donation

A donation entity represents the information coming with each donation. Each donation record includes the donor, the charity, amount, currency, status, and timestamp. We included other constraints for a donation record such as consent for the donation to be used for a service in a specific geographic region. The target region of the donation is represented with the coordinates specified in this record. This measure can help prevent the donation from being used outside of the intended destination. An expiry date attribute is also provided to mark the timeframe of the donation. Each donation must be spent before the provided expiry date, or the donation would be cancelled. This measure can help enforce the donation to be spent within the expected timelines instead of postponing.

3.3.4.2.3 Aid Handover

We model the delivery events as a series of handovers. For our aid delivery scenario, the Aid Handover structure is used to record these handover events. The aid materials that are going to a destination would be handed from one member to another in the blockchain system. The handover can be happening in person or it can happen autonomously using sensors. The record intends to store all possible information related to the actors involved in this event.

3.3.4.2.4 Delivery

Delivery records represent the delivery in the last mile. In our application, drones deliver the aid items to the destination coordinates. Delivery events such as state of the delivery and completion of the delivery are recorded to the blockchain. All these records create a trace of how a specific donation
is used. A delivery proof can be attached to a donation to further convince the stakeholders on the delivery of the aid. Drones add this proof of delivery to the delivery record.

3.3.4.2.5 Monitoring Events

Monitoring of the IoT events is a significant additional value provided by our blockchain. Delivery operations are a chain of handover events. Since the modern implementation of handovers and interactions involve a significant amount of IoT devices, a structure to record the observations is needed. We created an entity to record the observations from the monitoring devices. This entity contains attributes to record the key information on the delivery event and add the readings from the devices in a flexible structure where the type and value of the readings are provided.

3.4 Demonstration

As of May 2019, there were 861 active blockchains (BitDegree, 2019). Most of these blockchains were part of the cryptocurrency boom and created as specific products serving specific purposes. As the target products fail, the related blockchains also disappear (BitDegree, 2019). In contrast, some blockchains are developed as platforms, which provide infrastructure to people who would like to develop their own projects by utilizing underlying libraries and algorithms.

In order to design and test our application, we needed a blockchain platform. Among top blockchain platforms, Ethereum is the most popular while Hyperledger Fabric is the most popular permissioned blockchain. The world’s top enterprises openly support Hyperledger Fabric. We chose Hyperledger Fabric since Ethereum is a public blockchain system that is not suitable for the privacy requirements that may be required for our application. Hyperledger Fabric network consists of nodes that may have different roles. These blockchain nodes are called peers. All peers retain a copy of the ledger. Some peers are configured to run smart contracts. Others may be only focusing on transaction processing and commit operations.

3.4.1 Programming Language

Hyperledger Fabric provides alternatives for developing smart contracts. Go language is the default choice, and it is the natural language for the Hyperledger. Go was created with the intention of being faster than Python and not as complicated as Java. A Go system has great simplicity to be built and tested quickly. Go language has some useful features that are compliant with the philosophy of blockchain, for example the compiler errors are thrown for unused variables. Blockchain technology is a medium that distributes values and applications to several participants in the network and not wasting any memory is essential.

3.4.2 Number of Members

Each member of the blockchain is an independent server. Each member runs the blockchain software and communicates with the other members to conduct blockchain transactions. For the sake of limiting the related computational expenses, we limited members in the blockchain. We created a total of four peers: charities, drones, aid workers, and service providers. These four members are a good representation of four different roles in the blockchain application. Service providers are the only mandatory blockchain member.

3.4.3 Block Size

Block size is an essential factor in blockchain operations. Each block is created with a predefined number of transactions or a predefined time frame. If there is a high number of drones, there would be a high number of transactions created simultaneously. Large throughput of transactions is dependent on the block creation capacity.
3.4.4 Channels and Security

Channels are logical separators between the different lines of businesses. Each disaster would have its own channel separating the participants and rules from other disasters. Channels are the scope of security and business rules. For our implementation, we created a channel named “aid channel.” Users are registered, and roles are assigned on this specific channel.

3.4.5 Resources (CPU, Memory)

We used three computers in our experiment. The blockchain application is installed and configured on two Linux based computers while the application testing the system ran on a third Linux based computer. CPUs for these computers are Intel Core i7, and each has 2GB RAM.

3.4.6 Transaction Size

The transaction limit in the Hyperledger is 99 MB by default. Our record structure is minimal, and this limit is not a significant factor in our tests.

3.5 Evaluation

3.5.1 Scenario

There are several online disaster databases that keep detailed records of disasters. Public Safety Canada (PSC, 2021) has the Canadian Disaster Database that contains disaster information related to more than a thousand disasters. The types of disasters we want to simulate are where people are scattered to a geographical area and delivery for aid is needed. Hurricane and flood are practical examples of our scenario.

3.5.1.1 Delivery Targets

There are no specific and detailed datasets for the locations of the disaster victims at times of flooding and similar disasters. Some victims evacuate their homes, some victims move to higher grounds, some return to their homes, and many gather together to join forces against possible dangers. Due to the lack of readily usable data, we started from a superset, then identified the aid target area. Finally, we included all the addresses in the designated target area into our delivery destination list.

Ideal data for experimenting with the delivery assurance as part of disaster relief would be the data detailing where disaster victims are located during a major disaster. This data would show how victims are scattered, how they gather, where they wait for aid, and what the group demographics are. Such data would help us to accurately simulate the delivery of the care package to the right location with coordinates, deliver the precise amount of aid, and prevent any waste. However, such data do not exist. Without the coordinates of disaster victims and demographics, we have to use simulation in order to conduct our tests. Our simulation assumes that each address/house to which aid is sent is occupied. Our experiment is not sensitive to the number of people in the target location. The aid material that we are proposing to be delivered is enough for large crowds for long durations.

We need a superset of target locations to be used as delivery destinations. For this purpose, we use the Open Addresses dataset. Open addresses are a global dataset for addresses. Data in this dataset consists of a list of addresses with longitude, latitude, street number, street name and the city. Addresses are incomplete with missing elements such as postal code, but this dataset provides us with the minimum data that we need for conducting our simulation.

We created a relational database system (RDBS) in order to work on the addresses (see Figure 3). For this purpose, we created an MS SQL Server database in the Microsoft Azure environment. Our main table into which we loaded the delivery targets is called ADDRESSES. 525,545 Toronto addresses are loaded into this table. In order to load the data from its source (CSV files) to the SQL tables, we created a data ingestion pipeline using the Azure Data Factory.
Data cleansing is a big part of the analysis of big data systems. We continued with enriching and cleaning the data according to our planned tests. For our Toronto tests, we modelled a flood in the Toronto area. In order to simulate the rise of the water and calculate the impact area, we used elevation. The elevation attribute is not included in the Open Addresses dataset. To add this data to our dataset, we used JAWG\(^8\), which is an interactive map provider. We created an application that goes through the ADDRESSES table in the database, and for each record, calls the JAWG API to add the elevation data. The application code is provided on GitHub (the link is removed to protect the anonymity of authors).

In order to find a suitable set of destinations, we worked on a model that creates a limited number of destinations. Closure of the main roads is a factor in defining the flood area. When the main roads are closed, alternative delivery channels such as drone delivery can be used as a substitute. Our flood simulation took place as follows. We gradually increased the water levels and observed the impact of the flood on the Toronto map. Toronto is a city with variety of elevations where there are multiple high and low points in the city limits. When the waters were raised 180 meters, we observed a large area of land was still above water. Using Google Earth, we increased the water body around Toronto to 190 meters with the aim to limit the number of delivery destinations to a manageable figure and simulate the conditions that make conventional means of delivery not possible. This is a realistic scenario resembling relatively recent flood situations in Puerto Rico, Houston, and New Orleans to name a few cities in North America where the elevation difference between the highs and lows is less.

---

**Figure 3.**
Addresses table

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Condensed Type</th>
<th>Nullable</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddressId</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>LON</td>
<td>decimal(12, 9)</td>
<td>Yes</td>
</tr>
<tr>
<td>LAT</td>
<td>decimal(12, 9)</td>
<td>Yes</td>
</tr>
<tr>
<td>NUMBER</td>
<td>varchar(128)</td>
<td>Yes</td>
</tr>
<tr>
<td>STREET</td>
<td>varchar(128)</td>
<td>Yes</td>
</tr>
<tr>
<td>UNIT</td>
<td>varchar(128)</td>
<td>Yes</td>
</tr>
<tr>
<td>CITY</td>
<td>varchar(128)</td>
<td>Yes</td>
</tr>
<tr>
<td>DISTRICT</td>
<td>varchar(128)</td>
<td>Yes</td>
</tr>
<tr>
<td>REGION</td>
<td>varchar(128)</td>
<td>Yes</td>
</tr>
<tr>
<td>POSTCODE</td>
<td>varchar(128)</td>
<td>Yes</td>
</tr>
<tr>
<td>ID</td>
<td>varchar(128)</td>
<td>Yes</td>
</tr>
<tr>
<td>HASH</td>
<td>varchar(128)</td>
<td>Yes</td>
</tr>
<tr>
<td>Elevation</td>
<td>decimal(6, 2)</td>
<td>Yes</td>
</tr>
<tr>
<td>SimulatioId</td>
<td>int</td>
<td>Yes</td>
</tr>
</tbody>
</table>
As seen in Figure 4, this gradual increase left a smaller land that is suitable for our experiments. This figure also shows how the main roads are flooded and alternative disaster relief efforts are not possible. By querying our database, we found that 17,546 addresses in Toronto would be above water with this level of flooding. We plotted some of our delivery targets on to a map and validated the locations are mainly around a park, which has a small airport, hangars, storage locations and a large park. Table 1 lists the experimentation steps described above. Details of the per meter change in the number of above-water addresses are provided in Table 2.

3.5.1.2 Crisis Centers

Under normal conditions, we cannot operate drones from airports. However, previous flood examples indicate that the aid often accumulates in airports and needs to be distributed from there. We acknowledge the fact that drones cannot fly from the airport near the park in the target area, but the park itself can be the distribution and crisis center.

3.5.1.3 UAVs

UAV companies are service providers. They are members of the blockchain network, and handle blockchain-related tasks that are not handled by drones. They coordinate drones, schedule tasks, and manage the physical characteristics of the autonomous operations such as handling the loading of the drones or other mechanical maintenance tasks. In the simulation, we developed a java application to represent the UAV company. This Java application manages the scheduling and coordination of the UAVs.

In the simulation, each UAV is represented with a separate thread. Threads are executed simultaneously in the java virtual machine (JVM), and each thread receives its task from the UAV company. The UAV simulation application includes the majority of the simulation logic as the success and failure logic for the deliveries are coded in the UAV applications as drone-based factors. Drone-
based factors include drone failure such as a drone that fails and gets lost during the operations. Some drones are programmed to try to reach far away delivery points and cannot come back to the base successfully. The rate for failure is assumed to be 1%-3% (according to the details provided in the physical failures section below). Each drone picks a delivery task and marks it into the blockchain. During the delivery operation, it creates its IoT conditional monitoring records. When the delivery operation is completed, the UAV creates the delivery completion transaction. Finally, the UAV returns home, and the mission is accomplished.

The proposed methods related to drone delivery might face some challenges. In this section, we discuss these challenges and present our assumptions related to these inherent limitations. We also discuss possible available solutions to these problems (or solutions that are expected to be available in the near future).

3.5.1.3.1 Drone Operations: Regulations and Legislation

There are regulations related to operating Unmanned Aerial Vehicle (UAV)s. For example, in Canada, Canadian Aviation Regulations (CAR, 2021) list the rules for drones that are 25 kg and less. Small drones under 250 grams are reasonably free to use in line of sight with at least a five-meter distance from people. Certification is mandatory for advanced recreational drones that are heavier.

<table>
<thead>
<tr>
<th>Elevation Min (&gt;)</th>
<th>Elevation Max (&lt;=)</th>
<th>Number of Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>200m</td>
<td>…</td>
<td>516</td>
</tr>
<tr>
<td>190m</td>
<td>200m</td>
<td>17018</td>
</tr>
<tr>
<td>180m</td>
<td>190m</td>
<td>47279</td>
</tr>
<tr>
<td>170m</td>
<td>180m</td>
<td>52337</td>
</tr>
<tr>
<td>160m</td>
<td>170m</td>
<td>64408</td>
</tr>
<tr>
<td>150m</td>
<td>160m</td>
<td>60120</td>
</tr>
<tr>
<td>0m</td>
<td>150m</td>
<td>283855</td>
</tr>
</tbody>
</table>

Table 1.
Number of addresses for each 10 meters

<table>
<thead>
<tr>
<th>Elevation Min (&gt;)</th>
<th>Elevation Max (&lt;=)</th>
<th>Number of Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>195m</td>
<td>196m</td>
<td>1348</td>
</tr>
<tr>
<td>194m</td>
<td>195m</td>
<td>1558</td>
</tr>
<tr>
<td>193m</td>
<td>194m</td>
<td>2054</td>
</tr>
<tr>
<td>192m</td>
<td>193m</td>
<td>2739</td>
</tr>
<tr>
<td>191m</td>
<td>192m</td>
<td>3112</td>
</tr>
<tr>
<td>190m</td>
<td>191m</td>
<td>3770</td>
</tr>
<tr>
<td>189m</td>
<td>190m</td>
<td>3958</td>
</tr>
<tr>
<td>188m</td>
<td>189m</td>
<td>3888</td>
</tr>
<tr>
<td>187m</td>
<td>188m</td>
<td>4277</td>
</tr>
<tr>
<td>186m</td>
<td>187m</td>
<td>4567</td>
</tr>
<tr>
<td>185m</td>
<td>186m</td>
<td>4962</td>
</tr>
</tbody>
</table>

Table 2.
Number of addresses for each one meter
than 250 grams. For these advanced drones, rules are very restrictive, such as the maximum altitude of 90 meters, 75-meter minimum distance from people/buildings, maximum of 500 meters from the operator, and a minimum of 9.5 km from airports (Lem and Lem, 2019). Furthermore, each province has a trespass act governing the very likely conflict when a drone is using the airspace above a private property. Some law experts define trespassing as a mere presence of a drone on private property (McNabbon, 2019). The same sources indicate that the purpose of the drone is irrelevant when there is a conflict in trespassing. Altitude restrictions define the airspace of private property.

Autonomous flights are forbidden in most countries such as the UK. However, most restrictive countries have processes in place to issue a permit for significant size experimental autonomous delivery (Kobie, 2016). The list of publicly permissive countries was originally limited to Costa Rica, Iceland, Italy, Sweden, Norway, and UAE in 2017 (Jones, 2017). Recently, more and more countries are granting permits for proven vendors (Butterworth-Hayes, 2018).

3.5.1.3.2 Drone Delivery Problems

Drone delivery is not yet widely in practice due to some problems related to the nature of this activity. First, there is a weather challenge where drones are extra vulnerable due to extensive exposure. Cold weather severely degrades drones’ battery capacities. Fog, snow, and rain are also extreme challenges for drones (Enderle, 2019). Drones’ flight range is another current limitation (Matyszczyk, 2015). Delivery drones need to have extended capabilities to deliver items to long distances and return for re-load. It can be argued that in order to be economically viable, a drone must be able to complete multiple delivery flights with one charging of its battery. Replacing the batteries after each flight would delay the next take-off and decrease the total number of runs. A manual replacement is the default option, but it is also the most speed inefficient. If the battery replacement can be automated, the total number of deliveries would increase. Another concern is the safety of the drones. Drone delivery needs enhanced drones with price tags that are higher than recreational drones. These expensive pieces of equipment can be vulnerable to physical attacks, abuse including shooting, theft, or theft by finding. Connectivity is another challenge (Tuerk, 2019). A delivery drone needs to be connected for both the management and the safety of the delivery operations. Connectivity is key to normal operations. For mobile networks that enable such connectivity, 5G technology is ideal. Network providers see this opportunity to boost the utility of drones and have started preparing for it (Paul, 2019). With the superior qualities of 5G such as minimal latency and more than 100 times faster data speeds, drones are expected to provide several more services besides flying (Gemalto, 2016). Better coordination of drone fleets through takeoff and landing speeds up the operations.

While looking forward to a world with drones handling delivery operations, we are aware that the safety concerns must be handled, and potential issues must be resolved. These potential issues include accidents such as a drone hitting electric wires and causing a fire. Drones crashing into each other in the sky is possible with overcrowding and concentration of operations in certain areas. Drones becoming an obstacle to other air vehicles is the most common concern at the moment, but it is also possible that drones become a concern for land vehicles on the highway and cause accidents. When drones are loaded with deliveries, additional issues such as falling drones or packages may become a concern (Korman, 2019).

3.5.1.3.3 Assumptions to Address Drone-Related Challenges

We assume legislation will be changed towards enabling the technology, at least during disaster recovery for which legislators can justify the changes on flight rules. Disaster conditions need to be treated considering the realities of extraordinary circumstances. We assume drones are protected from harm where flight safety, including public safety, is achieved with technological enhancements. Even in disaster conditions there can be people targeting delivery drones in order to steal their load or for the salvage value of the drones and their parts. We consider such loss as part of the failure rates in our analysis.

We assume network technologies enable continuously connected drones to exchange data at seamless speeds with the introduction of 5G technologies. Continuous connectivity enables drones to
do continuous monitoring and real time interaction. However, even if the drones are not continuously connected, our blockchain system would still serve its purpose. There would be differences in the timeliness of the data where all communication would be postponed until the drone is returned to its base. Our experiments analyze both scenarios.

Existence of battery technologies that enable extended flight time and the long range of the drones are essential assumptions about the drone delivery experiments. In our analysis reported later in the paper, we present the calculations related to assumptions on the battery technology and range. In sum, since our proposed blockchain solution is flexible and the blockchain technology is in general open to change and improvements, the current assumptions do not constitute any permanent incompatibility with potential changes in the future.

3.5.2 Simulation Tests

We ran simulations and collected data in repeated sessions in order to understand the blockchain behavior and test the capacity of the blockchain network we constructed. During the tests, we assumed that all drones are clients for our system and are using the system by continuously issuing delivery event transactions. It is assumed that no time is wasted at the flight of delivery as the deliveries are instant. These tests reveal the ideal concurrent clients for our system, and we observe the maximum throughput we can reach with the current setup. We completed sixty tests in this category to record the impact of changes in the test variables towards the overall system performance.

3.5.2.1 Throughput

We ran our tests for each combination of the test variables: block size, block timeout, and the number of concurrent clients. For each combination, we simulated one thousand deliveries and collected the resultant data. We also repeated our tests with several higher numbers of loads and observed very comparable results.

3.5.2.1.1 Distribution with Block Size Set to Ten and Block Timeout Set to Two Seconds

In order to understand the impact of our test variables, our first major test was to accomplish the same amount of work with a different number of concurrent clients. We collected the change in the throughput value with respect to the changing number of concurrent clients. For the entire test, the block size is kept as ten, and the timeout is kept as two seconds.

Figure 5 shows the result of this experiment. It indicates that where the throughput in number of Transactions Per Second (TPS) is the performance benchmark, the lowest performance is observed where there is not enough activity. When there is only one drone issuing transactions, each transaction waits for the block to be completed. Since one client means one transaction, the transaction is packaged in a block only with the block timeout, which is 2 seconds. This behavior is observed even when there are nine clients, which are connected to the blockchain simultaneously, each issuing one transaction. Therefore, a total of nine transactions are created, and the system times out and completes the block before the next transaction.

When the number of concurrent clients reaches the block size, the behavior of the blockchain system changes and blocks are issued as soon as the block size is reached. Therefore, any number of concurrent clients above the block size results in good performance. The results show that the best performance is achieved when the number of concurrent clients is the same as the block size. When the number of concurrent clients exceeds this number, some clients are serviced with the current block immediately while others wait for the next block.

Figure 5 shows a decline in the performance with an increasing number of concurrent users after the number exceeds the block size. There are two reasons for such a decline. The first is the queue: when there are twenty clients trying to issue transactions at the same time, the first ten transactions are packaged in the same block, and the next ten transactions are packaged in the next block. Even though the decline in performance is not high, there is still an added wait time for the owners of the second set of ten transactions, which decreases performance. The second suspected reason performance degradation is the simulation software. Simulation of multiple concurrent clients is done using thread
3.5.2.1.2 Block Size vs. Increasing Number of Clients

In order to demonstrate the full impact of changes in the test variables, we show the performance of our system for each tested block size. Figures 6-9 show how the peak performance of the system for each block size change is related to the number of concurrent clients. For each network setting, the peak performance is accomplished when the block size is set to be the same value as the number of concurrent clients.

When the block size of the blockchain is set to 20 transactions, we observed the peak performance where 20 drones access the blockchain concurrently. The variation of the performance with respect to changing concurrent clients is shown in Figure 6.

When the block size of the blockchain is adjusted to 30 transactions, we observed the peak performance where 30 drones access the blockchain concurrently. The variation of the performance with respect to changing concurrent clients is shown in Figure 7.

Figure 8 and Figure 9 show that this observation is uniform: when the block size of the blockchain is adjusted to 40 transactions or 50 transactions, we observe the peak performance where number of drones accessing to the blockchain concurrently is equal to the number of transactions that form a block.

What we found in this analysis indicates that the blockchain systems are different from other client-interaction based systems such as web servers. In this simulation, we demonstrated the client delay and peak performance requirements for blockchain systems. Since block size and the block timeout parameters define the transaction completion for each client issuing their transactions in a time frame, the number of concurrent clients equal to the block size maximizes the throughput.

3.5.2.1.3 Distribution With Block Size Equal to the Number of Drones

Since peak performance is accomplished when the number of concurrent clients is the same as the number of transactions bundled in a block, we tested this configuration in different settings.
Figure 6.
Performance (TPS) with increasing concurrency (block size = 20)

Figure 7.
Performance (TPS) with increasing concurrency (block size = 30)
Figure 8.
Performance (TPS) with increasing concurrency (block size = 40)

![Graph showing TPS with increasing concurrency (block size = 40)]

Figure 9.
Performance (TPS) with increasing concurrency (block size = 50)

![Graph showing TPS with increasing concurrency (block size = 50)]
Figure 10 shows that the performance peaks where the number of concurrent clients and the block size is twenty. This type of peaks is often related to the infrastructure capabilities such as network and hardware. When the number of concurrent accesses exceeds the healthy capacity of the system, clients form queues and servers split their capacity between managing the queue and processing requests. Clogging of the system makes processors make high numbers of context switching, which slows down the processing and reduces the TPS performance.

3.5.2.1.4 Overall Distribution

Figure 11 presents a summary of the variety of the settings we tested as part of our load testing. The labels in the X-axis indicate the number of concurrent clients, block size and the timeout value for block creation. The value of ‘D’ represents the number of concurrent clients. The value of ‘B’ is the block size, and the value of ‘T’ is the block creation timeout value.

3.5.2.2 Latency

In addition to throughput, we measured several metrics including the latency for each concurrent client. Latency in blockchain transactions is the amount of time a client waits from sending a transaction to the blockchain until the successful completion of the transaction. Until a transaction is added to a block, our blockchain does not return a successful response. Figure 12 displays a summary of these values. Latency increases with the number of concurrent clients. Clients wait for more for each transaction when there is an increasing number of concurrent clients. Each client waits more than three times as long, on the average, in the 50 concurrent client scenario as in the 10 client scenario, and the maximum wait time in the 50 client scenario may be more than five times the maximum wait time when there are ten concurrent clients. We believe the simulation application also has an impact on this value to increase since 50 threads using the same set of resources would diminish the performance of the system.

In the drone flight scenario, each drone spending an increasing amount of time to conduct blockchain transactions is certainly not desired. However, considering the total amount of time it will take to load the drones and change the batteries in addition to the time each drone will spend flying, even the maximum value of ten seconds is not high. We analyzed the latency for our system to find out the configuration that would result in peak performance. Figure 13 displays latency changes with

![Figure 10. Performance (TPS) where the block size = concurrent clients](image)
the change of the number of concurrent clients. The increase in the number of concurrent clients shows an unusual behavior around the point where the number of concurrent clients is close to the block size (for this graph the block size is 10). However, overall, there is a steady increasing trend for the average, minimum and maximum values. As the blockchain network creates its blocks for the clients, more clients issuing transactions means more clients waiting. Two factors are essential in our load test. First, all clients continuously issue transactions, one immediately after the previous one is issued. Second, the simulation threads share resources and cycles from the same resources.

3.5.2.3 Results of the Tests

We calculated earlier that items need to be delivered to 17,546 different addresses for the aid distribution scenario. From the load testing perspective, we adopt this large number ignoring whether the physical delivery is possible or not. For these addresses, we run the blockchain simulation with its peak performance settings, i.e., twenty concurrent clients with a block size of twenty. Twenty concurrent clients can represent twenty drones delivering items in an instant, and immediately continue their subsequent deliveries without any flight, loading, battery replacement or downtime. The simulations are created to test the extremes in order to test the blockchain system under heavy load. In this scenario, completing all deliveries took 22 minutes and 43 seconds. Each transaction took 77.68 milliseconds, achieving 12.87 transactions per second. This performance indicates that if each drone would issue a blockchain transaction for each delivery, approximately 23 minutes of this operation will be spent on blockchain operations. These results show that the blockchain-related additional time cost is insignificant. Blockchain technology does not bring any additional cost to the overall system in terms of performance. With an assumption of each drone completing each delivery task in 10 minutes, the per delivery weight of a one message blockchain transaction is 0.012947%.
Issuing 77 blockchain transactions for each delivery would make the blockchain transaction time cost to increase to 1% of overall delivery.

In our scenario, the average distance to a delivery target is 8.2 km, hence the return trip is 16.4 km. With the assumption that each drone can fly 100 km/hr, completing each delivery task will average 4.8 minutes, and the per delivery weight of a one message blockchain transaction is 0.027%. Issuing 37 blockchain transactions for each delivery would make the blockchain transaction time cost to increase to 1% of overall delivery. These figures indicate that issuing aid delivery transactions on a blockchain network is feasible.

3.5.2.3.1 Performance Analysis Results

The result of the performance analysis is summarized Table 3. Analyzing the distances from our crisis center to the delivery targets, we calculated the average distance to delivery targets as 8.2 km. The farthest address is 18.494 km from the crisis center. The total distance to be flown is 143,339.831 km. This number is 3.58 times as long as the earth’s circumference. If each drone would make a single delivery at each time, the total distance increases to 286,680 km. With a 100 km/hr drone speed, this distance would translate to 2,868 hours of UAV flight. With 100 drones, our planned aid mission concludes in approximately 28.7 hours. Two hundred drones would reduce the time to approximately 14.4 hours. Since each UAV will fly approximately 4.8 minutes to serve the distance at the average distance, one drone would create one transaction every 4.8 minutes. One hundred drones will need a throughput of 0.35 TPS, and two hundred drones would need 0.69 TPS. In the most extreme scenario, 3,707 drones with one message per delivery would result in peak performance of 12.87 TPS.
Figure 13.
Latency change with growing number of clients

![Latency graph showing latency changes with growing number of drones.](image)

Table 3.
Summary of findings

<table>
<thead>
<tr>
<th>Experimental Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average distance of delivery targets</td>
<td>8,200 meters</td>
</tr>
<tr>
<td>The farthest address</td>
<td>18,494 meters</td>
</tr>
<tr>
<td>The total distance to be flown</td>
<td>143,339,831 meters</td>
</tr>
<tr>
<td>Number of deliveries at each mission</td>
<td>1</td>
</tr>
<tr>
<td>Drone speed</td>
<td>100 km/hr</td>
</tr>
<tr>
<td>Total flight hours</td>
<td>2,868 hours</td>
</tr>
<tr>
<td>Number of drones planned</td>
<td>200</td>
</tr>
<tr>
<td>Total time of mission</td>
<td>28.7 hrs with 100 drones</td>
</tr>
<tr>
<td>Performance requirement</td>
<td>0.35 TPS for 100 drones</td>
</tr>
<tr>
<td>Expected time for each transaction / drone</td>
<td>4.8 minutes</td>
</tr>
<tr>
<td>Maximum # of drones for our blockchain</td>
<td>3707 drones</td>
</tr>
</tbody>
</table>
3.5.2.3.2 Physical Failures

Reliability engineering literature indicates that complex components and system failures can be represented with a bathtub curve (Collins and Warr, 2018) as in Figure 14. These complex components and systems fail in greater rates at the beginning of their utilization as low-quality components with defects fail fast. After the initial usage, for a long period of usage, the failure rate is flat and at its lowest as failures are limited to random failures. In our simulations, we assume the initial quality control tests and initialization procedures eliminate the dead-on-arrival equipment. Therefore, we start our process with the low probability of having a failure. We increase the probability towards the wear-out period. (Collins and Warr, 2018).

Table 4 is a categorization of the addresses to receive delivery and their distances to the crisis centers. This table also includes possible failure ratios in order to estimate the number of failures. The literature includes a wide range of failure figures. For each UAV one failure is foreseen for each 1,000 hours of flight (Petritoli, Leccese, and Ciani, 2018). We translate this number to be a 0.1% failure rate for each one-hour flight. This is our random rate of failure in the bathtub curve. After 14 km of flight this probability increases exponentially as shown in Table 4. The percentage of failures is a factor in the total time of completion for all of the delivery jobs. Excessive failures can also result in failure of the overall delivery operation. However, the total number of failures has no impact on the blockchain load. Whether to activate new UAVs or to continue operations with fewer drones are operational decisions. The impact of these decisions on the blockchain system is either no change or a decrease in load. Table 4 estimates 70 failures throughout the entire delivery process. If there are 200 drones to start with, only 130 of them survive at the end of the mission. All of these figures are in acceptable ranges.

Figure 14.
Bathtub curve of the drone failures

<table>
<thead>
<tr>
<th>Distance (d) From Crisis Center</th>
<th># of Delivery Destinations</th>
<th>Estimated Failure Rate</th>
<th># of Failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>d&lt;12000</td>
<td>11459</td>
<td>0.1%</td>
<td>12</td>
</tr>
<tr>
<td>12000=&lt;d&lt;13000</td>
<td>304</td>
<td>0.1%</td>
<td>1</td>
</tr>
<tr>
<td>13000=&lt;d&lt;14000</td>
<td>363</td>
<td>0.1%</td>
<td>1</td>
</tr>
<tr>
<td>14000=&lt;d&lt;15000</td>
<td>736</td>
<td>0.2%</td>
<td>2</td>
</tr>
<tr>
<td>15000=&lt;d&lt;16000</td>
<td>546</td>
<td>0.4%</td>
<td>3</td>
</tr>
<tr>
<td>16000=&lt;d&lt;17000</td>
<td>2314</td>
<td>0.8%</td>
<td>19</td>
</tr>
<tr>
<td>17000=&lt;d&lt;18000</td>
<td>1687</td>
<td>1.6%</td>
<td>27</td>
</tr>
<tr>
<td>18000=&lt;d</td>
<td>137</td>
<td>3.2%</td>
<td>5</td>
</tr>
</tbody>
</table>
We also analyzed the impact of battery changes on the performance of the system. The battery change operations have no impact on the blockchain system performance. However, they can be considered a factor in calculating the total time it will take to complete the disaster relief. Table 5 details the analysis that indicates 5,586 battery changes would be needed until the end of the entire delivery operation, considering each battery supports only 100 km of flight. This number seems operationally and logistically challenging. We consider this challenge as the inherent challenge of drone flights. The total number of batteries to replace is proportional to the total flight distance. However, this factor does not have an impact on the blockchain solution.

3.6 Communication

As the different components of this solution were generated and tested, the resulting models and findings were presented at high impact academic conferences\(^{10}\).

A summary of the steps of this design science research is presented in Table 6.

4. DISCUSSION

In this paper, we report on the design of a blockchain-based aid delivery assurance system. We focus on disaster recovery and provide use cases showing disaster recovery is a suitable target for blockchain implementations. We detail the value that blockchain brings to disaster recovery efforts and services. As our aid delivery scenario uses autonomous vehicles, we define the value proposition of blockchain technology for the services provided by autonomous vehicles.

4.1 Theoretical Contributions

Our main theoretical contribution is the modeling of the delivery business as a blockchain and detailing the steps of this process. The role blockchains play in assuring delivery is detailed through our design process. This process can be adapted as a framework to other delivery scenarios where blockchain technology can be in the backend for the storage of records that assure delivery in an immutable system. Our simulations can also be adapted to the variations in the physical delivery events.

For the modeling of the delivery as a blockchain, we identified the supplies that are being delivered, the nodes in the network that they are delivered to/from, the delivery vehicles as the main components of the design that would be supported by a blockchain in the backend. As these are common elements in a generic supply chain as well as in a more specific case of aid delivery, the model can be adopted as a template for a blockchain for which we identified service request, donation,

<table>
<thead>
<tr>
<th>Distance (d) From Our Crisis Center</th>
<th># Delivery Destinations</th>
<th># Flights With a Single Battery</th>
<th># Battery Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>d&lt;12000</td>
<td>11459</td>
<td>4</td>
<td>2865</td>
</tr>
<tr>
<td>12000=&lt;d&lt;13000</td>
<td>304</td>
<td>3</td>
<td>102</td>
</tr>
<tr>
<td>13000=&lt;d&lt;14000</td>
<td>363</td>
<td>3</td>
<td>121</td>
</tr>
<tr>
<td>14000=&lt;d&lt;15000</td>
<td>736</td>
<td>3</td>
<td>246</td>
</tr>
<tr>
<td>15000=&lt;d&lt;16000</td>
<td>546</td>
<td>3</td>
<td>182</td>
</tr>
<tr>
<td>16000=&lt;d&lt;17000</td>
<td>2314</td>
<td>2</td>
<td>1157</td>
</tr>
<tr>
<td>17000=&lt;d&lt;18000</td>
<td>1687</td>
<td>2</td>
<td>844</td>
</tr>
<tr>
<td>18000=&lt;d</td>
<td>137</td>
<td>2</td>
<td>69</td>
</tr>
</tbody>
</table>
aid handover, delivery, and monitoring events as data entities used in transaction management. These entities are also applicable to various disaster aid management scenarios, therefore can be adopted for future blockchain based systems.

Our simulations were based on data that did not readily exist, which would be typical for the simulation of a disaster scenario where all the relevant variables would not be in a single database. The approach we took was to identify the relevant variables for the simulation models and integrate data from multiple databases on those variables. The incremental approach we present in Tables 1 and 2 can be adapted to other disaster scenarios by future research in aid delivery simulations with or without a blockchain backend.

4.2 Practical Contributions

We implemented the required blockchain solution using Hyperledger technology. We experimented with the ability of this solution to address the capacity and capability concerns. Our load tests and analysis results show that our blockchain functions adequately and has a high capacity in specific configurations. Our experiments showed that achieving high throughput is possible with integration of blockchain in aid delivery operations. We identified the configurations that provided the highest throughput. We also identified the latency values at each configuration. We concluded that blockchain performance is sufficient for running a large-scale aid delivery. We also showed that drone failures and battery changes do not pose a risk to the blockchain implementation.

Our experiments showed that achieving high throughput is possible with integration of blockchain in aid delivery operations. We identified the configurations that provided the highest throughput. We also showed that drone failures and battery changes do not pose a risk to the blockchain implementation.
we targeted. From a latency perspective, we showed that the latency we would introduce to the system is negligible compared to the duration of the physical operation. We also showed that UAV failures and battery changes do not pose a risk to the blockchain implementation.

Through this study, we observed that the performance metrics of the blockchain application are optimum when there is a stream of transactions coming to the blockchain platform that will not create a queue of clients, and that will not be too small compared to the expected flow. This performance peak can be accomplished by planning the delivery events in order to create a desired inflow of transactions. Since most of the variables for the flights such as distance and speed are known, the aim of the scheduling process would be eliminating peaks. If drones are assigned to tasks that would create events in a uniform distribution, the load on the system would be uniform and the performance would be maximum. A scheduler can be developed for blockchain applications that will schedule UAV flights and assign destinations to optimize the performance of the blockchain.

The most significant constraints identified are around the maturity of autonomous delivery and network connectivity. We conducted our experiments while addressing these concerns with realistic assumptions. Future research may build on our assumptions, test what if scenarios, or improve them where more precise data are available. Our work improves the aid delivery processes with the addition of assurance using the immutable records of blockchain technology. Blockchain technology is a recipe for cryptography-based trust injection over distributed records for multiple untrusting stakeholders. Our simulation studies validate the applicability of this proposed system. Further, the validation we received from an industry expert strongly suggests that this solution is feasible and applicable in industry.

5. CONCLUSION

Research and development can take a number of directions following our work. Drones surveying disaster scenes while delivering packages would be a great benefit to the stakeholders. Rescue teams, insurance companies, governments and the public can benefit from the extra information such surveillance would provide. This way, we would have an intelligently recreated map of the disaster area, which is usually different from the “pre-disaster” map. Destroyed bridges as well as newly formed rubble bridges are good examples of notable variations between the before and after disaster maps.

There are several potential improvements to our solution with improvements on decisions made related to path choices and related to dropping the load. With integration of artificial intelligence (AI), specifically pattern recognition capabilities into the system, drones can recognize the existence of people and target these people to deliver the aid. If there is no sign of people in the target drop zone, AI based decision algorithms hosted in the drone would choose to take the aid to an alternative drop target. Considering the disaster conditions are volatile and people would move to safer locations, this advanced decision making would make sure aid reaches people. Targeting also should avoid newly flooded areas. Some of these can only be recognized by the drone when it is on the scene. Future work of AI integration can improve targeting and rerouting.

Another future direction for our work is to create a framework that can guide the implementation of our disaster delivery blockchain solution to other cities. Disaster management offices in each city can plan for the recovery of the next possible disaster using our proposed solution as a basis. In our simulation, we identified an area with 17,546 addresses to be serviced with 200 drones in 14.4 hours. Similar levels of planning can identify the number of resources needed, necessary timelines to deliver the aid and geographic area to be covered for the next disaster recovery.

In this work, we used multithreading as part of the implementation. Although multithreading offers advantages to our system, it might cause some issues when sharing the CPU and memory. Lack of CPU and memory saturation suggests that the impact is not significant. However, in the future, we plan to have an implementation of the system that uses other alternatives such as implementing the experiment in a large distributed system with several compute units. In this way, we will prevent
resource sharing and possible added delays due to processes taking turns to utilize the CPU and memory.

The ideas and concepts we presented and tested in this research can be considered disruptive, especially to those who are responsible for public safety and thus management of disaster reliefs in various jurisdiction areas. This makes it important to consider the drivers of adoption of such solutions for stakeholders of such implementations including local authorities, donors, and taxpayers. Our research reported in this paper established the technical and functional robustness of the proposed solution with clear statement of assumptions and what if scenarios. Nevertheless, as suggested by recent literature (Schuetz and Venkatesh, 2020), blockchain based systems are no exception to the technology adoption processes and constraints. Therefore, an important potential direction for future research is to empirically test the perceived usefulness and acceptability of our solution in user studies above and beyond the small-scale interview we conducted with an industry expert. Given the ever-increasing frequency and severity of natural disasters and the feasibility of technology to circumvent disaster relief challenges, the natural next step in this important area is the identification of management challenges in the deployment of much needed state of the art solutions for people in urgent need.

ACKNOWLEDGMENT

This research was partially supported by research funds from the Department of Computer Science and Ted Rogers School of Management, Toronto Metropolitan University.
REFERENCES


BitDegree. (2019, May 15). *Did You Know There are 861 Blockchains?* Retrieved from https://blog.bitdegree.org/did-you-know-there-are-861-blockchains-c60e1720fad5


Tian, F. (2017). A Supply Chain Traceability System for Food Safety Based on HACCP, Blockchain and Internet of Things. 13th International Conference Service Systems and Service Management (ICSSSM), 1-6.


**ENDNOTES**

1. https://www.modum.io/
2. https://www.tradelens.com/
5. https://www.foliawater.com/foliafilterpapers
6. See for example: https://www.vaughan.ca/cityhall/emergency_planning/Pages/default.aspx
7. https://openaddresses.io/
10. The detailed citations are concealed for blind review purposes.
APPENDIX – DATA MODEL DETAILS

Service Request

type ServiceRequest struct {
    Status string `json:"status"
    ServiceType string `json:"servicetype"
    DestinationLongitude string `json:"destinationlongitude"
    DestinationLatitude string `json:"destinationlatitude"
    Timestamp string `json:"timestamp"
}

Donation

type Donation struct {
    Donor string `json:"donor"
    Charity string `json:"charity"
    Amount int `json:"amount"
    Currency string `json:"currency"
    Status string `json:"status"
    DestinationLongitudeStart string `json:"destinationlongitudestart"
    DestinationLatitudeStart string `json:"destinationlatitudestart"
    DestinationLongitudeEnd string `json:"destinationlongitudeend"
    DestinationLatitudeEnd string `json:"destinationlatitudeend"
    Timestamp string `json:"timestamp"
    Expiry string `json:"expiry"
}

Aid Handover

type AidHandover struct {
    Timestamp string `json:"timestamp"
    DestinationLongitude string `json:"destinationlongitude"
    DestinationLatitude string `json:"destinationlatitude"
    AidItemId string `json:"aiditemid"
    FromPrincipal string `json:"fromprincipal"
    FromAgent string `json:"fromagent"
    FromSensorHost string `json:"fromsensorhost"
    FromSensor string `json:"fromsensor"
    ToPrincipal string `json:"toprincipal"
    ToAgent string `json:"toagent"
    ToSensorHost string `json:"tosensorhost"
    ToSensor string `json:"tosensor"
    Donation string `json:"donation"
}

Delivery

type Delivery struct {
    Status string `json:"status"
    ServiceType string `json:"servicetype"
    DestinationLongitude string `json:"destinationlongitude"
    DestinationLatitude string `json:"destinationlatitude"
Monitoring Events

type DeliveryMonitoring struct {
    DestinationLongitude string `json:"destinationlongitude"
    DestinationLatitude string `json:"destinationlatitude"
    Timestamp string `json:"timestamp"
    AidItemID string `json:"aiditemid"
    DeliveryPrincipal string `json:"deliveryprincipal"
    DeliveryAgent string `json:"deliveryagent"
    DeliverySensorHost string `json:"deliverysensorhost"
    CurrentLongitude string `json:"currentlongitude"
    CurrentLatitude string `json:"currentlatitude"
    OtherMonitoringType string `json:"othermonitoringtype"
    OtherMonitoringValue string `json:"othermonitoringvalue"
}
Mehmet Demir is a solution architect working for Amazon Web Services. Dr Demir received his B.S. and M.S. degrees in Computer Engineering from the Bogazici University, Istanbul, Turkey in 2001. He holds an MBA from Schulich School of Business, York University Toronto Canada, in 2007. He received his PhD from Computer Science at Toronto Metropolitan University (formerly Ryerson University), Toronto, Canada, in 2020. Besides his academic studies, Dr. Demir has over 20 years of industry experience as an enterprise and solution architect in some of the most prestigious Canadian companies in consulting, banking, health, retail, telecommunications and energy. He also has several certifications from Amazon (AWS), Microsoft (Azure), IBM, Oracle, TOGAF, PMI-PMP, ITIL, Scrum.org and BEA Systems related to his technical, architectural and leadership work.

Ozgur Turetken is a professor at the Ted Rogers School of Management at Toronto Metropolitan University (formerly Ryerson University) where he has been serving as associate dean for research since July 2019. He is one of the past chairs of the AIS Special Interest Group on Decision Support and Analytics (SIG DSA). His research interests are in business analytics and human computer interaction with an emphasis on modeling and testing of text mining and information presentation methods. His previous work has appeared in journals such as ACM Database, AIS Transactions on HCI, Communications of the ACM, Communications of the AIS, Decision Support Systems, IEEE Computer; Information & Management, Information Systems, Information Systems Frontiers, and MIS Quarterly Executive. Dr. Turetken holds a BS in EE, an MBA (both from Middle East Technical University – Ankara, Turkey), and a PhD in Management Science and Information Systems (Oklahoma State University).

Alex Ferworn is a Professor of Computer Science in the Faculty of Science at Toronto Metropolitan University (formerly Ryerson University). Alex earned his PhD from the U of Waterloo, his MSc from the U of Guelph and his B. Tech from Toronto Metropolitan University (formerly Ryerson University). Previous experiences include working extensively within telecommunications and finance industries as well as serving as an infantry officer in the Canadian Forces Reserve. He is the Graduate Program Director of Ryerson’s Master of Digital Media program as well as the Director of the Network-Centric Applied Research Team (N-CART) lab. Alex has a broad range of research interests which inevitably have something to do with sensing and decision support using dogs and/or robots working in disasters or emergency situations. Having worked extensively with Urban Search and Rescue (USAR) teams in Canada and the United States, he seeks to address the needs of emergency First Responders and managers working in difficult environments. More recently, he has worked with the Chemical, Biological, Radiological, Nuclear, and high yield Explosives (CBRNE) teams within several police services to explore the use of serious games for training and operational purposes. His award-winning work has been widely publicized in the media. In 2013 he was named the EURAXESS Canadian “Science Slam” champion for his ability to communicate complex ideas to a general audience in compelling ways. “Partners In Research” named him their Canadian “Technology Ambassador” of 2014 for his body of work and his outreach activities. In 2019 his innovative work in finding lost and wandering patients with dementia was featured at the CRAM Learning Festival.

Mehdi Kargar is an Assistant Professor at Ted Rogers School of Information Technology Management at Toronto Metropolitan University (formerly Ryerson University) and an adjunct assistant professor at the Department of Management Sciences at the University of Waterloo (since 2017). He is also a member of SOSCIP Scientific Advisory Committee representing Ryerson University (since 2019). From 2016 to 2018, he was an Assistant Professor of Computer Science at the University of Windsor. His research concerns data management, business intelligence, big data analytics, and sustainability. He is awarded multiple research grants since 2017, including NSERC DG and SOSCIP TalentEdge. He received his Ph.D. in Computer Science from York University (2009-2013). He is specifically interested in designing and developing search systems over enterprise databases for business users. His research is published in data management top-tier venues like VLDBJ, PVLDB, ACM SIGMOD, IEEE ICDE, IEEE TKDE, IP&M, ACM CIKM and SIAM SDM. He holds a M.Sc. and a B.Sc. in Software Engineering from Sharif University of Technology in Iran and was ranked 7th in Iranian National Scientific Olympiad for university students in Computer Engineering in 2006.