Blockchain Based Smart Contract Framework for Demand to Supply Matching In Transport and Logistics

H.M.N.T.P Herath

Blockchain Based Smart Contract Framework for Demand to Supply Matching In Transport and Logistics

H.M.N.T.P Herath
Index No: 14000466

Supervisor: Dr. T.N.K de Zoyza

January 2019

Submitted in partial fulfillment of the requirements of the
B.Sc. in Computer Science Final Year Project (SCS4124)

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: H.M.N.T.P Herath

……

Signature of Candidate Date: 08/01/2019

This is to certify that this dissertation is based on the work of Mr. H.M.N.T.P Herath under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Principle Supervisor's Name: Dr. T.N.K De Zoyza

……

Signature of Principle Supervisor Date: 08/01/2019

Co- Supervisor's Name: Mr. P.A Walpita

……

Signature of Co-Supervisor Date: 08/01/2019

Abstract

Transport and logistics industry is one of the large markets in the world economy. This industry has multiple modes of operations such as local haulage, air freight. And ocean freight. The systems that are currently operational uses third party service providers to establish trust between stakeholders with paid compensations. It requires considerable amount of effort to manage these independent silos of systems with high cost.

This dissertation concentrates on Proof of concept framework for the transport and logistics industry with goal of eliminating trusted third party services. The proposed framework is based on blockchain and driven by smart contracts. The framework uses data from existing monitoring systems in the transport and logistics industry to create a new way of performing transactions where human interaction is minimized. The proposed framework tries formulate transport contract agreement into a smart contract and execute it according to the data obtained from IoT networks **(**data oracles).

A trust between two stakeholders can be established using a private blockchain and Smart contract protocol. The framework is capable of creating smart contracts between two parties for the given set of inputs and conditions. Smart contract protocol escrows the transport job fee as proof of stake. Data from data oracles are used identify state changes in Smart contract where it performs transactions when agreed conditions are met. Transport agreement violation penalties are embedded into the Smart contract where it resolves such incidents without bias to either stakeholder.

Preface

Third party services that are used to establish trust between multiple parties provide their service at rates. These services are a necessary requirement in current execution models used in the transport and logistics industry. Involvement in these service providers extends the time taken to process transportation and logistics services.

The proposed framework provides a novel approach to mitigate the problem of expensive third-party service providers by proposing a method to establish trust between supplying and demanding parties in transport and logistics. It formulates the transport agreement into a Smart contract. Smart contract application in transport and logistics domain is a novel where existing studies only use blockchain 1.0 technologies. The use of existing monitoring systems as data oracles is a new idea where data is used for Smart contract condition validation. This framework applies off-chain computation layer design and off-chain data store designs newly for transport and logistics industry. The prototype which is based on blockchain and driven by Smart contract is an application created to test the framework. This is a novel software application for transport and logistics context.

Acknowledgement

I would like to express my sincere gratitude to my research supervisor, T.N.K De Zoyza, senior lecturer of University of Colombo School of Computing and my research co-supervisors, Mr. P.A Walpita and Mr. T.G.A.S.M. De Silva for providing me continuous guidance and supervision throughout the research.

I would also like to extend my sincere gratitude to Dr. M. G. N. A. S. Fernando, senior lecturer of University of Colombo School of Computing and Mr. G.P.Seneviratne, senior lecturer of University of Colombo School of Computing for providing feedback on my research proposal and interim evaluation to improve my study. I also take the opportunity to acknowledge the assistance provided by Dr. H.E.M.H.B.Ekanayakeas the final year computer science project coordinator.

I appreciate the feedback and support provided by my friend Dulaj Sanjaya and others to achieve my research goals. This thesis is also dedicated to my loving family who has been an immense support to me throughout this journey of life. It is a great pleasure for me to acknowledge the assistance and contribution of all the people who helped me to successfully complete my research.

Table of Contents

List of Figures

List of Acronyms

IoT – Internet of Things

ABI – Application Binary Interface

ETH – Ether (Ethereum Currency Notation)

API – Application Programming Interface

EVM – Ethereum Virtual Machin

KYC – Know Your Customer

HTTPS – Hyper Text Transfer Protocol Secure

TLS - Transport Layer Security

Chapter 1

Introduction

1.1.1 Background to Research

Transport and logistics industry is one of the largest industries in the world. In the year 2015 along it generated revenue of 8.1 trillion dollars and expected to grow to be a market that will generate 15.5 trillion dollars in revenue by the year 2023 [1]. Air freight, ocean freight and ground freight are main three models that are being used in current transport and logistics. These transport models involve multiple parties to complete a transport job from initial transport job agreement to final delivery of cargo. In this long process, it involves a lot of documents, systems, and personnel to guarantee that transportation of cargo is done under agreed terms. Where transport and logistics provider can be identified as the supplier and service requesting party can be identified as the demanding party

Transport job agreement is a document that is currently being used to specify and form the agreement between the two parties that take part in this process. This document contains all the necessary instructions to complete a transport job. It contains granular level information like cargo pickup locations and order of cargo pickup, a route to follow, time and date of cargo pickup and delivery, final delivery locations and drop-off order of cargo, payment schemes and other conditions like fumigation, a temperature range of cargo if cargo is temperature sensitive. Letter of credit is another document that is being used in this process which deals with the payment details and acts as the guarantee of the payment according to payment scheme that supplying and demanding parties have agreed to. Letter of credit is a document that is processed via banks of supplying and demanding parties. Expensive services from an external or third party (a bank) are introduced to this process in order to establish trust between two parties and payment upon completion of transport job according to transport job agreement.

Currently, there are systems that operate in real time, is being used to monitor every aspect of a transport job from cargo pickup to cargo drop off [14] [15]. These systems use IoT networks and web portals as data input mechanisms. Such systems monitor the progress of a transport job, current condition and checks those values against the transport job agreement in order to identify any violations have been done so far. Current systems provide visibility throughout the process of transportation into a detailed level and those are provided and maintained by supplying party and used by both supplying and demanding parties. These systems play a vital role in the daily activities of transport supplier and provided to demanding parties by supplying party as a value-added service in most cases.

1.2. Research Problem, Research Questions

1.2.1 Research Question

How to introduce and formulate Smart Contracts and its execution for transport and logistics industry and mitigate the problem of altering Smart Contracts due to real time data from oracles?

1.2.2 Sub Questions

In order to find answers to above mentioned research question it was broken into several sub questions so that steps and tasks to the final goal will be clearer in the implementation and design stages.

- Identifying and selecting suitable conditions from the transport contract agreement and formulating Smart Contract(s).
- How to create data oracles and connect them with Smart contracts
- How to performing an initial coin offering

By creating a working prototype of the proposed framework answers to above mentioned questions was found.

1.2.3 Research Significance

In real world document trail and data from live tracking systems are two different systems that are not connected. But by connecting these two systems using blockchain and Smart contracts a complete autonomous system can be created. This framework will lead to reduce number of hard copy documents that is already needed in the process.

A trusted third party is needed in the current process to ensure trust between two parties. This is a service provided by banks, but at very high service charge. In a simple scenario at least two banks are needed in order to achieve expected level of trust. In this framework, it eliminates this trusted third party using blockchain, internal

cryptocurrency. This would have a great impact on the current way of doing business, since almost all business models and current systems are oriented around banks. When this centralization is removed by introducing blockchain it will be much faster and economical way to carry out transport and logistics as a company.

This framework uses Smart contracts, which reflects conditions in the transport job agreement. Using Smart contracts and the data from data oracles the monitoring of a transport job can be fully automated. In this no supervision would be needed since the data from the multiple oracles along the transport job path will be used to evaluate the conditions and calculate the progress of a transport job. This will result in a platform were once a transport job is started, the required monitoring is minimal saving overall cost for a transport job.

Even though it is out of scope for this study, a Decentralized Autonomous Organization can be created on top proposed framework. Which will have no central authority, low cost organization. A marketplace might be good example for this scenario were demanding parties can create transport job and upload them to a marketplace where supplier can select and start working on the job while any kind of supervision will not be required to identify contract violation or job completion.

1.3. Project Aims and Objectives

Main aim of this research is to create a framework for transport and logistics industry which reflects the transport job agreement between demanding and supplying parties using Smart Contracts with Smart Contract creation and Smart Contract auto liquidation. Following aims can be identified when developing such framework.

- The framework will be able create Smart Contract depending on the agreement between demanding and supplying parties.
- Tries to find way to Model a financial transaction into smart contract that has a single auto liquidation or to smart contracts that has multiple auto liquidating points.
- Attempt to find a solution to problem in transaction execution when the Smart Contract is void so that neither supplier nor demand will make a profit or loss

In the below section identified objectives and how those objectives will be achieved are stated.

Long term goal of this project can be stated as, creating a fully automated organization that operates of its own without any central controlling authority or thrusted third party. Therefore this framework can be considered as a first step towards such organization which is called Decentralized Autonomous Organization.

Filling the research gap is one of the objectives in this project. As stated above identified research gap is considered and this project is designed in way so it covers most of the research gap as possible.

One objective is to eliminate users that impersonating legitimate users. To achieve this a permissioned private blockchain will be used. The users will be registered or added to platform after a KYC phase and verifying that each and every user is legitimate users.

Identifying conditions of transport contract agreement and formulating them using Smart contract is one of the main objectives of this project.

In this registration process the initial coin offering will be done to each legitimate user so that they have an initial balance with them. This is one objective of this project that is to propose a method to perform an initial coin offering and study its effects on the proposed system.

To use the data from existing monitoring systems and connect them with the framework is an objective because as stated data from these monitoring systems are already available and does not require large effort to access them. The hard part of achieving this objective is connecting these data with Smart Contracts.

Handing violations of transport contract agreement is a objective of this framework. In order to achieve this studies into Smart contract linking will be done in order to discover the possibilities of implementation such solutions practically.

1.4 Justification for research

Blockchain is a decentralized and distributed digital ledger of transactions which was cryptographically enabled and tamper-proof. It was the underlying technology for many cryptocurrencies starting from Bitcoin. Blockchain technology have played a key role in development of modern frameworks for financial services given its key features such as disintermediation, confidentiality and robustness. With the introduction of Ethereum in blockchain 2.0 it provided a platform where financial contracts can be directly brought in to the domain of computer science. This was greatly boosted by the new smart contract concept which works as the backbone of Ethereum. Smart contracts can be defined as event and state driven autonomous programs that runs on a blockchain to administer its assets. Furthermore smart contracts can be treated as scripts that are stored in a blockchain. Smart contracts are new being studied and implemented in researches in many areas such as Token Systems, Financial Derivatives, Online Voting, Decentralized Governance, etc. Mainly these applications can be categorized into three branches as financial applications, semi-financial applications and other applications

Researches and studies into introducing blockchain platforms are still being done in research labs and institutions around the world. Most of these studies use the blockchain as a distributed ledger and as an enabler of transparency. In context of transport and logistics industry introducing blockchain is still at very early stage where proposed and implemented frameworks are running in testing or early stages of deployment. For instance Maersk Line, being largest and conglomerate company in transport and logistics industry joined with IBM for a venture to introduce blockchain aiming to improve the cost of transportation and minimize lack of visibility and inefficiencies with paper-based processes in January 2018[16]. Even though there are projects or rather startups that are in operational level which uses blockchain such as Quasa[12] and ShipChain[17] but their functionality is still at questionable level since these platforms are not 100% functional yet. Implementing and introducing Smart contract frameworks to transport and logistics is novel idea that still at research level and very few researches have been done in this area. Even though there are studies that shown that Smart contracts can be introduced and implemented in transport and logistics [6],[9]. The potential of Smart contracts have been identified and documented

outside from the research arena form different transport and logistics institutes worldwide [11], [18].

Even though Smart contracts have been used in above mentioned researches and frameworks none of them use data from live tracking systems as data oracles (streams of input data) to validate and verify conditions in the Smart contract. A clear mapping has not been done between Smart contracts and Transport contract agreement. Because of that data from IoT networks (or Real time tracking systems) is not used executing Smart contracts. As mentioned above since transport and logistics is a large industry framework created by this research can be used to create autonomous, low cost platform for transport and logistics industry which does not have a trusted third party.

This research is focused on semi-financial system: Using blockchain and smart contracts in logistics and transportation industry. To narrow down the study area protocols in cargo shipment is considered in this research. Where a demanding party will make request to supplier containing number of conditions such as date of cargo pickup, date of cargo drop-off, route to collect cargo from warehouses and most importantly a payment scheme. The payment scheme defines how the payments will be made during the points of cargo pickup or finally at cargo drop-off. Once this requested is accepted by the supplier and agreed upon both parties have a contract between them. To capture data about ongoing transportation network of IoT devices and web portals are currently being used. Smart Contracts can be applied to such situations so that once the agreed contract is completed the transaction between two parties can be done without any trusted third party like a bank which will lead to save time and cost for both parties. The existing operations use letter of credit as a proof-of-stake in this framework it uses features of Smart Contracts as proof-of-stake. To facilitate Smart Contracts in such system a blockchain will be is used as distributed ledger and Smart contracts as mechanism to execute financial transfers. This research is more focused on behavior of auto liquidation feature in smart contracts which eliminates that thrust third party in supply and demand service of transport and logistics.

1.5. Methodology

After conducting literature review to study and identify scope existing systems, frameworks that uses blockchain for this type of applications it is understood that studies into this type of practical frameworks are less. Even though there are promising studies [9] that discover possibilities that exists with current technologies. This study will first review existing studies that uses blockchain technologies in transport and logistics applications or frameworks.

In the first phase of this study we will try to introduce and formulate smart contracts so that a transport agreement can be created and stored in blockchain. Auto liquidation of smart contracts will be addressed in this first phase using IoT devices as data oracles. Data oracles will be accessed from the smart contract, for this purpose existing frameworks such as Oraclize[24], and Zap[23] is studied to find the best solution to provide accessibility of data from outside networks since data oracles resides outside from blockchain network that smart contracts are stored on. Here it is assumed that the data and the data oracles are to be trusted and does not provide any false data. In order to increase number of points of verification (conditions as in transport agreement) multiple data oracles is used in a single smart contract. When using and accessing oracle data via API calls this study will use already available methodologies to create secure API endpoints and will assume that existing method of API security methods provide will provide sufficient security so that it will not be a point of security failure. This phase will have a evaluation at the end of the implementation of the framework which is supposed to reveal its overall performance.

The second phase of this study will be done on top of the resulting framework of the first phase and implementation will use the framework that is created in the first phase of this study. In the second phase of this study we will try to solve the problem of smart contract violations which will occur when agreed conditions of the smart contracts are not met in the smart contracts. To solve this problem we will mainly study on methods that can be used to link smart contracts together. When linked smart contracts will be able to see each other's state and communicate with each other by message passing. This step will start by trying out the architecture proposed in [21] as a solution to this. In order to stop either party from getting any unexpected profit or

loss in such situations creation secondary set of smart contracts with initial price variation will be studied as well. At the end of this phase an evaluation will be done to finalize and find the overall performance and the practicality of the resulting framework in this study.

In evaluation stages the framework will tested against practical scenarios as much as possible. Evaluation is done in order to evaluate applicability of this framework to real world transport and logistics operations in business use case perspective.

1.6 Outline of Dissertation

This dissertation follows structure as, in the second chapter existing literature is reviewed under several areas that are relevant to this study. The third chapter presents the proposed research design and the methodology followed by implementation demonstration of the proposed framework in the fourth chapter. Chapter five consists evaluation model and the results of this study. In final sixth chapter concludes this dissertation with the possible future works

1.7. Project Scope, Delimitations and Assumptions

1.7.1 Project Scope

Main objective of this research is to propose a framework for transport and logistics sector using blockchain and smart contracts. The research will try to address the issue that occur when such implementation is done in real world scenario.

Following components will be addressed in this research.

- User registration, Initial Coin Offering
- Transport job creation between supplying, demanding parties and creating smart contract(s) accordingly.
- Connecting data from IoT networks and web portals into Smart contracts.
- Smart contract execution, identifying Smart contract violation.
- Maintaining transaction privacy and transaction rate.
- Upon successful transport job completion, Supplier will be debited and demanding party will be credited.
- If transport job is not completed as agreed (when Smart contract violation occurs) new Smart contract will be created to carry out penalties.

1.7.2 Project Delimitations

This framework deals with a real world application. Therefore it is important to understand boundaries of this framework so that it can be studied and implemented within given time period. Following Components will be considered as out of the scope for this framework.

- Implementation and evaluation of air and ocean freight applications of this framework.
- Handling cargos which has less than a container load
- Limiting and manipulating value fluctuations of cryptocurrencies.

1.7.3 Assumptions

For developing this framework following assumptions will be made

- People use this framework act truthfully and does not provide false data.
- Data from IoT networks and web portals (data oracles) are accurate and correct
- Once a transport job starts no amendments will be done to transport contract agreement

1.8 Summary

Interactions in real-world transport and logistics industry involve many parties. This study focuses on the interaction between transport service provider and demanding party. This process mainly uses set of documents and most of the details regarding the transport job is stated in transport contract agreement document. In order to establish trust between above mention parties third party trust providers like banks involve in this transactions which increases transport cost. At the same time transport service providers use real time IoT networks to monitor transport job progress. Blockchain technology and its features can be used establish trust in peer to peer transactions while eliminating the need for a trusted third party. Smart contract can be identified as an enabler to formulate, create transport contract agreements. The research objectives mainly focus on creating a framework where trust and agreement conditions and be done. This framework will be evaluated using selected scenario from real world to emphasize on it practical applicability.

Chapter 2

Literature Review

2.1. Introduction

This chapter presents existing related work for this study. Blockhain and smart contracts provides an infrastructure to establish trust. Smart contracts which resides on the blockchain, reflects the real world contracts. Smart contract auto liquidation provides way to formulate agreements that involve currency transaction upon fulfilling defined conditions. In section 2.7 and 2.8 use of blockchain and smart contracts and financial frameworks in transport and logistic sector is reviewed with existing studies and solutions.

2.2 Blockchain

Blockchain is a distributed immutable ledger that is capable of storing transactions as data blocks. Non repudiation in blockchain network is achieved by using PKI when implementing applications using blockchain. Lately blockchain is used not only on cryptocurrencies but in many other sectors such as legal document preservation, healthcare data, IoT systems etc...[25]. since blockchain provides thrust mechanism it is suited can be used for online payment systems as well. Lately many blockchain implementations were done such as Rinkeby, Testnet. These networks are publicly open global blockchain networks that are open to anyone. There are private blockchain like Bankchain that can be accessed from anywhere in the world after valid authentication. For private setups in blockchain frameworks such as Hyperledger, Truffle can be used and depending on the requirement those can be made globally available or locally available (local machine setup). Even blockchain technology along (which is called Blockchain 1.0) provides way to implement immutability and non-repudiation it was not programmable, this led to limitation in use of blockchain in

research and application fronts. To solve this problem smart contracts were used in Ethereum [5].

2.3 Smart Contracts

The idea of Smart Contracts was originally described by the computer scientist and the cryptographer Nick Szabo real implementation and its value was discovered very recently after introduction of Bitcoin and blockchain technology [2]. Smart Contracts made its recognition when it was used as a part of the Ethereum cryptocurrency as the way of executing transactions between pre agreed parties [3]. Smart Contract is an autonomous agent which is stored in the blockchain. Smart Contracts can stored on the blockchain or off the blockchain. For off chain Smart Contracts those are migrated to blockchain once the execution is needed. Smart Contract hold an amount of virtual coins, some sort of private storage and its own predefined executable code. This properties of the Smart Contract is useful when it is used to represent a predefined agreement between two parties.

The code of the Ethereum contract is in a low-level, stack-based bytecode that is referred to as Ethereum Virtual Machine Code [4]. The EVM allows contract functions to have local state, while the contracts may have global variables stored on the blockchain. Contracts can invoke other contracts via message calls; outputs of these calls, considered to be a part of the same transaction, are returned to the caller during the runtime. Importantly, calls are also used to send Ether to other contracts and non-contract addresses. The balance of a contract can be read by anyone, but is only updated via calls from other contracts and externally initiated transactions [5]. There are different implementation of ethereum that can be used for smart contract implementation in languages but mostly used language is Solidity which is contract oriented high level language.

2.4 Payment Systems with Blockchain and Smart Contracts

In order to implement a payment system for services online not only digital currency is needed. In such system there should have a methodology to guarantee that the agreed service is provided before the transaction is made [1]. To achieve this level of guarantee trusted third parties are being used with additional cost. Smart Contracts can be used in order to eliminate the trusted third party in such transactions. By agreeing to a Smart Contract which is stored in a blockchain this elimination of trusted parties can be achieved.

2.5 Data Oracles

Smart contracts are stored in a blockchain as stated above. Therefore smart contracts cannot reach data which are outside to blockchain network. This isolation from real world data limits blockchain and smart contracts practicality. In order to avoid this isolation data oracles are used. Data oracle is third party data feed service designed to feed data into smart contracts that resides on blockchain. In the study carried out in [22] it states clear architecture that is has used data orcles with smart contracts that are created between two parties. Even though this study focuses more on how blockchain can be used as a software connector in this study the researchers provide implementation details about two use cases were they have used validation oracles to access and connect between two blockchain based applications.

The study in [26] propose a framework for authenticated data feed for smart contracts. In this framework the researchers which combines a blockchain front end with a trusted hardware back end to scrape HTTPS enabled websites and serve source-authenticated data to relying smart contracts. Another data feed framework is Oracalize [24] which achieves distributed trust by using a second service called TLSnotary [27], which digitally signs TLS session data. As a result, unlike TC which can flexibly tailor datagrams, Oraclize.it must serve data verbatim from a web session or API call.

2.6 IoT Networks for Transport and Logistics

Transportation and logistics industry has many aspects and vertices. This includes resulting products of a manufacturing process that needs to be moved from origin to destination using a transport and logistics chain. Such process used to individual atomic items (logistics objects) or collection of those item that will be processed as consignment (logistics unit load). For these types different types of IoT sensors is used varying from QR codes scanners, RFID tags which is be attached with logistics objects to GPS data feeding modules which is attached to logistics unit loads or consignments.

In traditional approach for transport and logistics lot of documents and human and inhuman verifiers are used. This approach suffers form lot of latency, expensiveness and leads to inefficient outcomes. In order to mitigate these problems IoT networks are incorporated with transport and logistics systems and methodologies. In the study [30] it is analyzed the effect of introducing IoT devices for supply chain management systems for manufacturing. Since transport and logistics sector serves as an integral part of and ERP system it implies that using IoT system for transportation will result in gaining better resource utilization and competitive advantage.

2.7 Blockchain for Transport and Logistics

With the rise of the breakthrough technologies as mention above financial sector is undergoing some major improvements in terms of transaction privacy and transaction execution methodologies. Particularly in transport and logistics sector there are only few researches have been undertaken to introduce blockchain and smart contract concepts. In the research [6] demurrage and maritime use case is studied. In situation of demurrage there will be many challenges to be faces such as complexity of shipping process, use of paper and small invoices. These challenges make cash collection of demurrage claims more complex. But in this study the main objective was to improve cash flow recoverability of the post-transaction expenses. Here the researchers propose a system where both parties in the agreement must validate the trigger event for smart contract auto liquidation. Auto liquidation is occurred when all the agreed terms are fulfilled according to the contract. But this study does not propose a method to auto generate or calculate demurrage claims using Smart Contracts. Furthermore this study limits its scope to demurrage use case and at the same time it does not tries

to calculate this claims using the Smart Contracts. In [7] the researches have limited their research to a perfect scenario where no delays or any disruption is done to transportation flow between supplier and demand, demurrage is taken place with the shipper and a third party other than demanding party.

The current system in transport and logistics is supply and demand business model where the demanding party specifies the requirements to a supplier and upon agreement to the necessary terms the service is provided. Such systems will use IoT [7] network to communicate the status of the transport job progress. Initial agreement terms and the above mentioned real time IoT data is used to generate smart contracts which is stored in a blockchain and to auto liquidate them. Every other scenario than the perfect scenario will result a void smart contract as explained in introduction section. This is a situation where existing systems or proposed systems do not have a proper solution.

Furthermore few studies have been done on how blockchain concept can be used in transport and logistic sector. When using blockchain for financial or government institution a private blockchain is recommended due to less openness and transaction and increased speed over a public blockchain.

Private permissioned blockchains are also heavily contested among blockchain enthusiasts because “opaque” blockchains that are limited to few known transaction processors (nodes) and limited access to users undermines the very concept of decentralization [8]. Furthermore in report it elaborates a complete use cases set where transparency and effectiveness of supply chain can be increased and Scaling Internet of Things and Digital Transportation Assets. In that report it explains more on how electronic wallets can be used in such framework systems.

In [9] it explains how an auditable trail of information can be created in asset tracking system which uses IoT and blockchain technologies. Rather than having multiple databases across multiple entities (stakeholders), if all the entities can use blockchain network that is set up to keep track of assets means that there is only one shared database need to keep track of. In explorative study carried out in [10] it states that global logistic industry is flawed and to achieve greater transparency blockchain can

be used. Lack of ability to trace the origins of assets in such systems is identified here as well. Furthermore the complexity of such system when using blockchain will be increased and it should be identified as enabler not as a negative entity which brings only complexity.

2.8 Smart Contracts for Transport and Logistics

Studies into use of smart contracts in transport and logistics is still at its inception. In the [11] it has identified use cases in financial derivatives and supply chain sectors. It states that by using smart contacts traceability to granular level inventories, delivering assurance and simplification of multi-party delivery can be achieved. To achieve this trusted oracles must be used in such a system. In [12] proposed system uses Smart Contracts in a transport and logistics platform. It states that claims for void Smarts Contracts can be manually created by its users. This paper does not reveal any implementation or further details about the proposed system other that from a very high level.

The Smart Contract implication in transport and logistics industry is still not well studied. At the same time problems that arise when undertaking such implication is not properly identified. On the other hand blockchain and IoT based blockchain studies and solutions are currently being studied up to this date. A gap in the above mentioned areas can be clearly identified and the studies in to those areas can be conducted with existing knowledge from the studies that are carried up to now.

2.9 Summary

This chapter begins with providing detailed review of blockchain and smart contracts. Blockchain technology is developed and implemented in many frameworks to use its features. Smart contract facilitate blockchain frameworks with the introduction of programmability and with the ability to perform auto liquidation. Data oracles removes the isolation by providing necessary interfaces to provide data. IoT networks with real time data will serve as data oracles. In section 2.7 and 2.8 presents existing studies or frameworks on introducing blockchain and smart contracts to transport and logistics sector.

Chapter 3

Design

3.1 Introduction

This chapter presents deign of the proposed solution to the research problem. It contains four main sections high level research design, blockchain network, and smart contract formulation, off chain database and data oracles. Above sections illustrates how the research design will achieve the intended aim of this research with the selected approach.

3.2 High Level Research Design

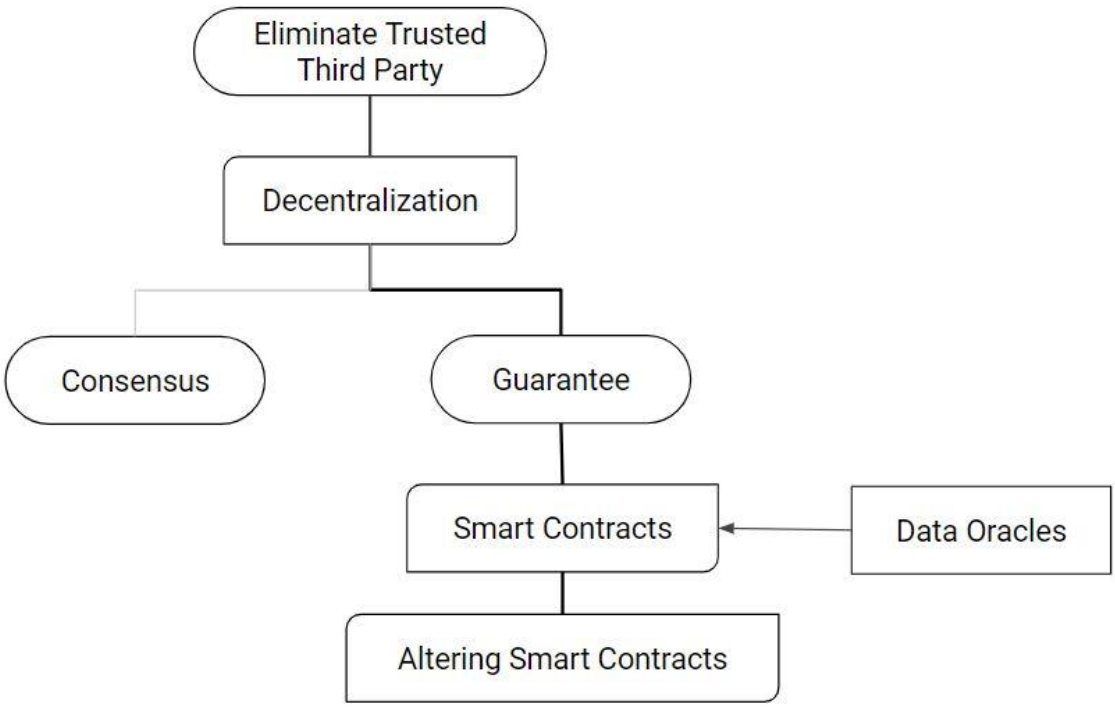


Figure 3.1 – High Level Research Design

Above diagram illustrates how the research is designed where the design decisions are selecting blockchain, formulating smart contracts, connecting real world data to smart contract and finally manipulating or avoid null smart contract by altering the smart contracts.

3.3 Blockchain Network

Removing trusted third party is one of the main concern taken in to consideration when creating design to this study. The central authority in transport and logistics industry is the service providers that is used to establish trust. This service providers is considered as central authority since they hold the authorization power to both parties. In order to do so decentralization approach is selected. This decentralization step removes central authority from this framework at the same time. To perform decentralization blockchain implementation is selected after the literature review. When using a blockchain as a base to framework byzantine fault tolerance dependability arises. In this framework anonymity of the users is not a concern since this framework is for use in transport and logistics sector where all the users will be identified and required to reveal the true identity. The blockchain selected in this as a facilitator because of its decentralization, non-repudiation and immutability features. Since blockchain has many implementations like BitCoin and Ethereum etc... and each has its own unique features. Ethereum blockchain is selected because Ethereum blockchain is less power consuming and it has ability to use with smart contracts is selection reasons. Private blockchain is the option when designing the blockchain as this study does not require to maintain anonymity of the users. By selecting a private blockchain probability for occur byzantine faults for minimized to negligible level since all the identities are revealed. If an adversary user was identified with in the network the users are able to communicate that off the framework and avoid making transactions with the identity.

Ganache blockchain simulator is selected form Truffle Suite and used to create a local private blockchain and account creation. This simulator provides mock blockchain accounts with specified amount of Ethereum for each account. It provides mining controls, blockchain log output to which is benefit for further analysis of the blockchain. To0 access the blockchain and perform operations Web3-eth library is used. This library provides the functionality to query for existing account addresses,

query blockchain statues and deploy smart contracts to the blockchain with specified contract state variable values.

3.4 Smart Contract Formulation

To formulate transport contract agreement smart contracts are created with limited number of verification points. For this study location and time, temperature verification parameters are used. Guarantee between two parties are ensure the nature of smart contract execution. At the smart contract creation stage following parameters are taken in as state variables.

- Supplier's blockchain address
- Demander's blockchain address
- Amount of Ethereum for fee
- Following for each way point and starting and drop off locations
    - Location name and unique id
    - Deadline to reach location (time in epoch)
- Unique id of the assigned vehicle
- Barcode Identifier Value
- Minimum temperature to maintain while delivery
- Penalty for delay delivery
- Penalty for not maintained agreed temperature in cargo

A smart contract is designed to reflect transport contract agreement in this study. It is designed in a way that the smart contract store only state variables and condition threshold values. It will be created and deployed to the blockchain for each transport contract agreement. All contract violation state identification logic is embedded into the smart contact. Set of state variables are declared in this Smart contract. These state variables are used when auto-liquidation is performed to identify and calculate penalty value.

Smart contract creation is designed to be performed via a web interface. All the values for the parameters is captured using this web interface. Once the smart contract is created it cannot be altered and it will be deployed to the blockchain. For this raw

smart contract are complied with the provided state variables. This compilation and deployment outputs contract deployed address and application binary interface (ABI). ABI acts as an interface for further interaction with that smart contract such as method invocation.

In this smart contract creation process the defined transport fee will be credited (deduced) by the defined amount and that amount is escrowed to the specific smart contract. This is an important transaction in terms of establishing guarantee. Since this transaction is between demander's account and the smart contract, it reflects that demander is a legitimate demander and willing to pay the agreed amount upon contract completion. This transaction revoke the ability of the demander to windrow from the agreement or avoid payment upon final delivery of cargo because the smart contract possess defined amount.

In case of initial contract terms violation the second smart contract is executed. In order to perform this first smart contract will pass its escrowed amount of ETH to the second contract along with the demander's address and the supplier's address. The second contract transfers the penalty amount back to demander's address and supplier's account will be added only with the amount that is available after applying the penalty.

To connect smart contracts with real time data two methods were found as viable options. First approach was to repeatedly query real time data from data oracles. This method was implemented in the Smart contract. This approach was not feasible since in order to perform operations smart contract consumes gas. This gas is taken from the value of the smart contract. Therefore this sort of a design will keep smart contract self-decreasing its value. Next method is minimize operations of the smart contract by revoking its methods only when a vehicle reached a defined location in the smart contract and pass location information and time stamp of the event to the smart contract for validation. This design requires an off-chain data store and off load data processing from the smart contract.

3.5 Off Chain Data

Smart Contracts models the state of the system, therefore, it is possible to store all the data on Smart Contracts. The main problem arises with this design is that even though the Smart Contracts models the state of the system is it expensive to store all the data on it. Dealing with live data from data oracles in Smart Contracts via method calls and message passing uses Gas. A system with such design will we flawed with the Gas leak that will eventually reduce Ethereum value of the Smart Contract by a significant amount.

To perform computation in a Smart contract it uses small amount of Ether. This amount is called Gas in Ethereum protocol. This value is a fraction of Ether therefore it is expressed by Wei (1 Wei = 10^{-18} Ether). Minimizing the number of computations is necessary to hold Ethereum value of Smart Contract. Important design decision is to store data that is required for computations as off-chain to perform calculations off-chain. Distance to the next location has to be calculated to check the vehicle has reached the location within the allocated time. This calculation needed to be performed for each GPS data item from data oracle. Since there are over a thousand GPS data items per transport job this is not possible to perform in Smart Contract without significant loss of Ethereum value.

Smart Contract should be minimal in size and number of computations that are performed inside. To achieve these two goals design of the off-chain data store is necessary. Summary of the circumstances that Smart Contract models the transport and its final states, Smart Contract holds state transitions that are cheap, checking whether there is a state transition is expensive. As suggested in [] Challenge Response Pattern is used to design the off-chain data store. State checking performed off-chain and the final result is communicated to the Smart Contract.

Off chain data store is designed in this study to persist the detailed information that is not stored in the smart contracts. This persistence layer is important in order to keep smart contract size minimum as much as possible. This off-chain data will act as

persistent layer for data from data oracles as well. For this off-chain data store MongoDb is selected after considering its ability to cope with changing data schemas.

Following main data defines a deployed smart contract, and are persisted after its successful deployment on blockchain.

- Unique id for transport contract agreement
- Address of smart contract
- ABI of the smart contract

This two values is used to invoke contract validation method that resides inside the smart contract.

3.6 Data Oracles

Establishing trust between supplier and demander while excluding the third party is one of the main intentions of this studies. One of the responsibilities of the third party is to monitor the progress of the transport job and verify it with agreed terms. To completely remove the third party a mechanism is needed to perform the monitoring. Data oracles provide multiple points of verifications for a given transport job. Using multiple points for verification is essential. It ensures the integrity of the transport job because it is more difficult to falsify multiple data entities than few data entities. Connecting existing monitoring networks as data oracles to Smart Contracts is identified as a solution to this problem. GPS data and data from barcode readers are used in this study as data from data oracles. As points verifications in this study location (GPS data) and contents of the cargo (Barcode) are selected.

Data oracles captures the relevant vehicle or cargo id and the measured value from its sensor. An IoT network with location and temperature is used in study as data oracles. Data oracles persist their data in off-chain data store. This design technique is uses intermediate service which part of off-chain data to perform calculations on real time data and extract information about status of the vehicle. To data oracles blockchain network and its operations are completely invisible.

3.7 Initial Coin Offering

Ethereum blockchain allows creating of custom cryptocurrencies and tokens that can be purchased using Ethereum. ERC20 is a protocol specifies the behaviour of such tokens. Tokens that are created with ERC20 protocol are compatible with other cryptocurrency platforms where native blockchain is not required for its operations. ERC20 protocol is written using smart contracts. ERC20 tokens can be transferred from one account to another like Ethereum.

In order to onboard supplying and demanding parties to the system, Initial Coin Offering is performed using ERC20 tokens. In this study, we propose a new token that uses ERC20 protocol that can be used in crowdfunding. Crowdfunding process that is proposed in this framework consists of following steps. Initially, the individual should reveal his identity to confirm he is a legitimate user. After providing identification information the individual should transfer Ethereum to the organization and then the organization will transfer ERC20 tokens to that individual.

By performing crowdfunding new suppliers and demanders (user) introduced into the framework while ensuring they are legitimate individual parties. This coin offering procedure is designed in a way that the identity of the stakeholders are revealed in very early stage since this framework is expected to be implemented within the legalities for transport and logistics in a given continent.

3.8 Summary

This chapter presented a detailed description of the research design and the approach used. This approach consists of blockchain setup, smart contract formulation, off-chain data store and data oracles. Main focus of this chapter to emphasize the design of smart contracts which serves as the main component in this study. Other sections include important design decisions and mechanisms that are proposed in this framework.

Chapter 4

Implementation

4.1 Introduction

This chapter provides a detailed description about implementation of the proposed final framework. Section 4.2 presents the software libraries and tools that is used in implementation. Section 4.3 presents high level description about the implementation architecture of the proposed framework followed by smart contract and NodeJs server implementation details.

4.2 Software tools and libraries used

Proposed framework designed with a private Ethereum blockchain. Ganache blockchain environment is used as a blockchain simulator for this study where real ETH is not required for perform transactions. This platform allows auto mining feature where actions of mining nodes are simulated. Ganache blockchain platform allows the configure mining to set block times to defined value. To develop smart contracts Solidity language is used in its 0.4.2 version. Solidity is a type-safe language where smart contract interaction is provided via ABI after deployment. React.js is selected as web platform base which is directly compatible with Web3.js library. Web3.js library is used to perform actions with the blockchain and smart contracts. Node.js web server is selected to host the web application. MongoDB no-sql data base serves as the off-chain data store in this framework.

4.3 High Level Implementation Architecture

This framework is implemented user two servers that runs independently. A NodeJs web server and a NodeJs backend API server. User interacts with the reactJs application interface where contract creation and deployment is performed. This ReactJs web application uses web3.js to retrieve blockchain and smart contract status information. NodeJs server and API server uses a common NodeJs database. Data oracles post its data to API server where data persistence is done. Upon receiving data from data API server perform operation to identify the relevant smart contract for received data.

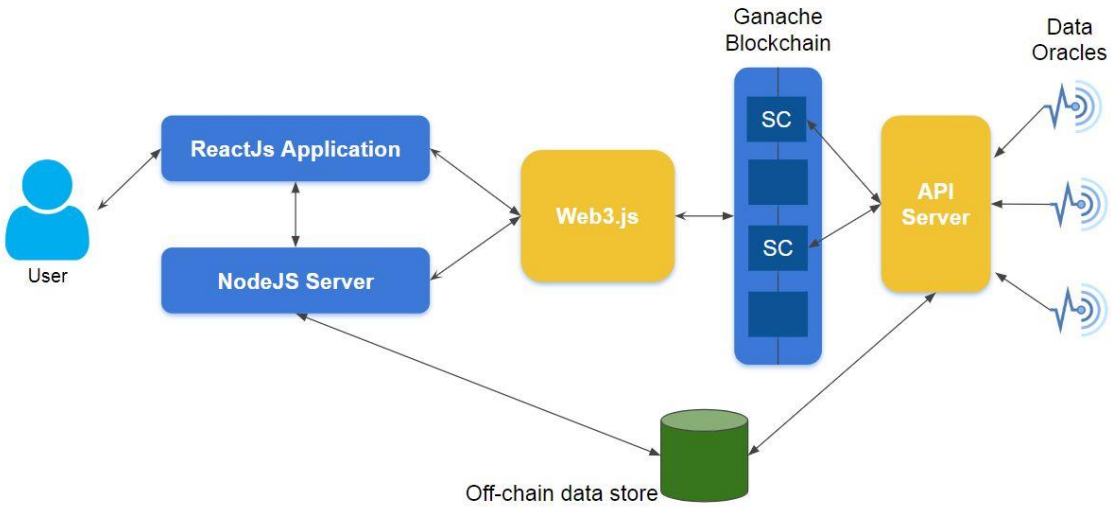


Figure 4.1 – High Level Implementation Architecture

4.4 Smart Contracts

```
contract TransportAgreement {

    uint public pickupDateTime;
    uint public wayPointDateTime;
    uint public dropOffDateTime;
    string public vehicalId;
    uint public conractValue;
    address public supplier;
    address public demander;
    uint public pickUpLocationId;
    uint public wayPointId;
    uint public dropOffLocationId;
    bool public isPickUpComplete = false;
    bool public isWaypointComple = false;
    bool public isDropoffComplete = false;

    constructor (address _supplier, address _demander, uint _pickupDateTime,
                 uint _pickUpLocationId, uint _wayPointDateTime, uint _wayPointId,
                 uint _dropOffDateTime, uint _dropOffLocationId, string _vehicalId,
                 uint _conractValue) public payable{
        supplier = _supplier;
        demander = _demander;
        pickupDateTime = _pickupDateTime;
        wayPointDateTime = _wayPointDateTime;
        dropOffDateTime = _dropOffDateTime;
        vehicalId = _vehicalId;
        conractValue = _conractValue;
        pickUpLocationId = _pickUpLocationId;
        wayPointId = _wayPointId;
        dropOffLocationId = _dropOffLocationId;

    }
```

Figure 4.2 – State variables and constructor of Transport Agreement Smart Contract

Smart contracts are used to embed the transaction logic into the proposed framework. The smart contract receives its values for state variables via the contractor mothed and maintain, change its states according to the values of state variables.

The code segment in figure 4.2 contract contractor is defined as a payable to allow this contract to receive ETH form regular transactions. This is used to escrow demander's payment amount to the smart contract when the smart contract is created. The contractor method is made public so it can be accessed from the outside world.

```
function updateContractState(uint dateTimeVal, uint locationId) public {
    if(locationId == pickUpLocationId
        && isPickUpComplete == false){
        if (pickupDateTime >= dateTimeVal){
            isPickUpComplete = true;
            approve();

        }
    }
    else if(locationId == wayPointId
            && isWaypointComple == false
            && isPickUpComplete == true){
        if (wayPointDateTime >= dateTimeVal){
            isWaypointComple = true;
            approve();

        }
    }else if(locationId == dropOffLocationId
            && isDropoffComplete == false
            && isWaypointComple == true
            && isPickUpComplete == true){
        if (dropOffDateTime>= dateTimeVal){
            isDropoffComplete = true;
            approve();
        }
    }
}
```

Figure 4.3 – Smart Contract method to update its state

The method shown in figure 4.3 is called from the API server when data from defined vehicle indicate that it has reached a location on its route. With the provided location id the smart contract identifies the type of the location as starting, waypoint or drop-off. And checks if this vehicle has reached the location before the agreed time. Times in this framework uses the epoch time format which uses integer to represent time because other comparisons use more gas to be executed. The order of the locations reached location in transport job is important since it is a condition in the initial agreement to identify the contract violation. In this validation process of the data smart contract checks the intended route was taken by this vehicle by checking flags of the prior locations are true. If all the conditions are adhered by the demander this smart contract will transfer agreed amount to the demanders address. This will finalize the transaction and no trigger from outside which presents the advantage of using smart contracts required.

```
function approve() public payable{
    if(isPickUpComplete && isWaypointComple && isDropoffComplete && isBarcodeValid){
        supplier.transfer(this.balance);
    }
    else if(isPickUpComplete && isWaypointComple && isDropoffComplete && !isBarcodeValid){
        demander.transfer(this.balance);
    }
}

function approveWithPenalty() public payable{
    if(isPickUpComplete && isWaypointComple && isDropoffComplete && isBarcodeValid){
        supplier.transfer(this.balance - totalPenalty);
        demander.transfer(totalPenalty);
    }
}
```

Figure 4.4 – Auto-liquidation methods

The figure 4.4 presents the code that approves the final transaction in initial contract. In case of the created contract violation is identified, smart contract will execute second function. Calculated penalty is deducted from the escrowed value in Smart contract. The balance value is transferred to supplier's account and penalty amount is returned back to demander's account.

4.5 Off-Chain Computation Layer

Off-chain computation layer is implemented using a NodeJs server. This layer performs mainly two actions.

- Compiling and deploying Smart Contracts.
- Process data from data oracles to identify state changes in transport job.
- Invoke Smart contract method via method call when state changes in a transport job.

NodeJs server in the web application is responsible for compiling (figure 5.1) smart contacts and deploying (figure 5.2) them in the connected blockchain and performing computations on data from data oracles. This compilation occurs in real time since the values for state variables are captured as the user inputs in the web application. For the compilation of Smart Contracts SolC library is used. Next step is migrating Smart Contracts into the blockchain. For this web3 library act as a connector between the Ganache blockchain and the server. After the successful migration of smart contract, this server persists ABI and the contact address in the off-chain database to be used in future for performing contact calls. All input data for smart contract creation is also stored in the off-chain data store as well.

```
var createContract = function(){
    const sourcePath = path.resolve(__dirname,'./contracts','TransportAgreement.sol');
    const source = fs.readFileSync(sourcePath,'utf8');
    return(solc.compile(source,1).contracts[':TransportAgreement']);
}

module.exports = createContract;
```

Figure 4.5 – Usage of SolC compiler

```
result = await new web3.eth.Contract(JSON.parse(interface))
        .deploy({data: '0x'+ bytecode, arguments: [supplier, demander,
                                                   startDateTime, startLocId,
                                                   wayPointDateTime, waypointId ,
                                                   endDateTime, endLoctonId,
                                                   vehicalId, value]})
        .send({gas: '1000000', from : demander , value: valInWEI }, function(error, transactionHash){
            console.error('error :', error);
            console.log('transactionHash :', transactionHash);
        });
```

Figure 4.6 – Deploying Smart contract into blockchain

Data oracles send their data to this layer. Location data is used to confirm whether a particular vehicle has reached a location that is specified in the transport job agreement. A geo-fencing method is used to obtain this confirmation. When a GPS data (latitude and longitude) is received, the great-circle distance between current position and next stopping location is calculated. Haversine formula (fig []) is used to calculate the great circle distance. If the calculated distance is less than the defined limit it is considered that the vehicle has reached the stopping location. When a vehicle has reached a location the relevant smart contract is called to update its state via a method call. This method call contains location identity and time which is needed to perform agreement condition verification in the smart contract.

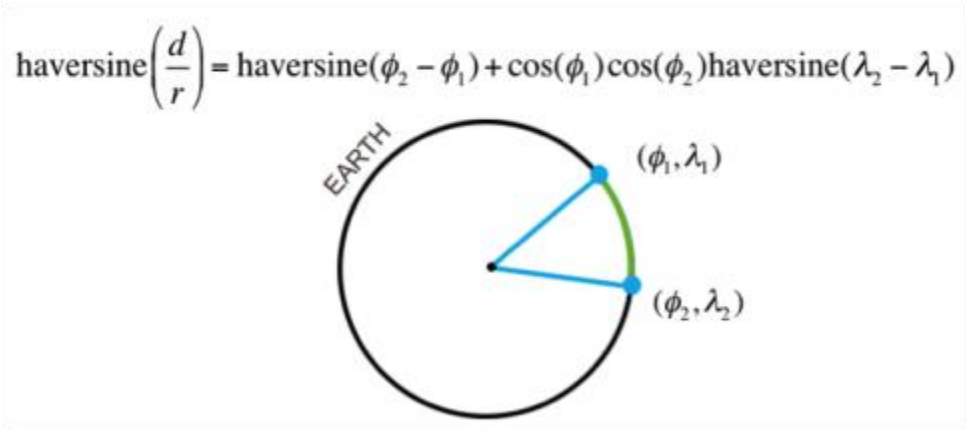


Figure 4.7 - Haversine formula used to calculate distance

4.6 ERC20 Tokens and Initial Coin Offering

The created token for this study is TranSmart Coin which uses TSC and its symbol. Token definition and its behaviour are specified in a smart contract. This smart contract contains the following definitions and functionalities.

- It stores the token symbol which will be is used currency exchanges.
- Defines total supply of toke that is in existence.
- Contains a mapping between accounts and its token balance.
- Implements "transfer" function that allows users to transfer token from one account to another.
- It implements an approve function that allows another account to spend tokens, like on a cryptocurrency exchange.
- It implements a “transferFrom” that allows another account to transfer tokens.

```
function transferFrom(address _from, address _to, uint256 _value) public returns (bool success) {
    require(_value <= balanceOf[_from]);
    require(_value <= allowance[_from][msg.sender]);

    balanceOf[_from] -= _value;
    balanceOf[_to] += _value;

    allowance[_from][msg.sender] -= _value;

    Transfer(_from, _to, _value);

    return true;
}
```

Figure 4.8 – Method used to transfer tokens between two accounts

A separate smart contract is created to perform the initial coin offering. This smart contract allows suppliers and demanders to purchase tokens. This purchase will make those parties stakeholders in the framework. The crowdfunding smart contract contains the following definitions and actions.

- It holds the address of the account that initiates the crowdfunding event as an admin account.
- Stores the token price
- Stores the number of tokens that is sold.
- It implements a "buyTokens" function that allows users to purchase tokens in the crowdfunding event.

- It implements an "endSale" function that allows an admin to end the crowd sale and collect the Ether funds that was raised during the sale

```
function buyTokens(uint256 _numberOfTokens) public payable {
    require(msg.value == multiply(_numberOfTokens, tokenPrice));
    require(tokenContract.balanceOf(this) >= _numberOfTokens);
    require(tokenContract.transfer(msg.sender, _numberOfTokens));

    tokensSold += _numberOfTokens;

    Sell(msg.sender, _numberOfTokens);
}

function endSale() public {
    require(msg.sender == admin);
    require(tokenContract.transfer(admin, tokenContract.balanceOf(this)));

    // Just transfer the balance to the admin
    admin.transfer(address(this).balance);
}
```

Figure 4.8 – Methods used for token crowd sale

4.7 Summary

This chapter states the software tools and libraries that is used to implement proposed framework. Underling functionality in each module of the framework was elaborated using code level descriptions. The behavior of the smart contracts in the framework is mainly focused in this chapter followed by the descriptions of the methods used for smart contract compilation and deployment. This chapter describes the technique that is used to handle smart contract violation within the framework. The final section states the mechanisms that are used to create ERC20 token and perform crowdfunding event to onboard users to the framework.

Chapter 5

Results and Evaluation

5.1 Introduction

This chapter elaborates the results from the execution of proposed framework. Since this study is a proof of concept for introducing smart contract for transport and logistics industry. There is no generalized method for evaluation. Evaluation is done by comparing the expected results and produced results from the framework. Section 5.2 presents the results of the implemented frameworks. All accounts that are used testing had 100 ETH as initial balance in evaluation.

5.2 Evaluation Method

Limitation of generalized evaluation models for proof of concept studies in blockchain and smart contract study areas is a drawback identified while performing the evaluation. Evaluation of this proof of concept framework done using multiple stakeholder accounts and stored data from IoT networks. The evaluation is performed in two directions as follows.

- Approach 1 - Evaluating the framework for different data inputs and observe the output.
- Approach 2 - Quantitative analysis of smart contract creation and contract method calls regarding Gas amounts spent.

The first approach to evaluation is done in two phases. In the first phase, data is manually fed into the framework only for state change events. Meaning that the data is fed to the framework only when a vehicle has reached a location in the transport job. The second phase, five datasets were used. Data is fed to the framework from transport throughout the entire transport job. These datasets consist of GPS data, Barcode

scanner data and Temperature data. In this phase, data was fed into the framework using a JSON file via API calls. To automate this a separate program written in Java was used. Below are the maps depicting GPS dataset**.**

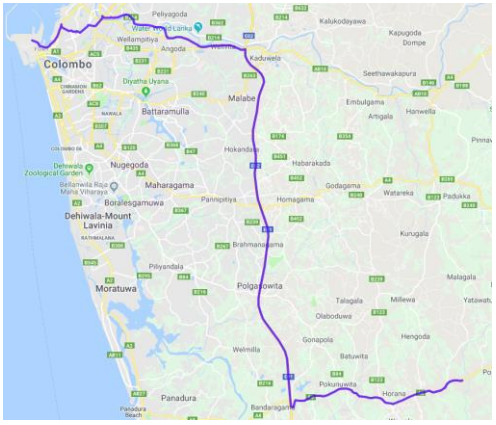


Figure 5.1 – Route followed from Colombo Fort to Ingiriya Bodyline Garments

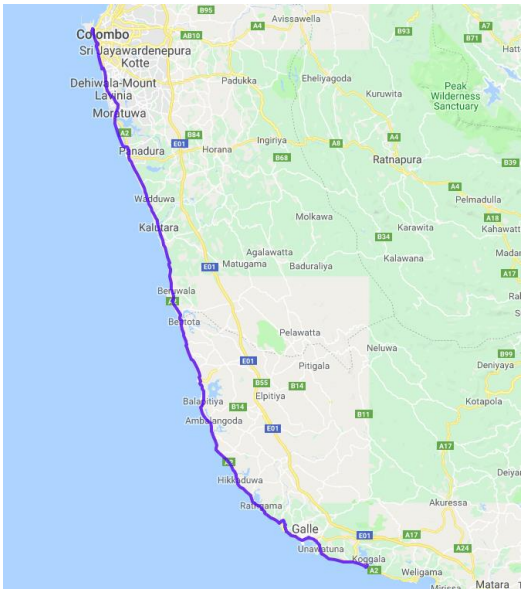


Figure 5.2 – Route followed from Colombo Port to Koggala MAS Via Kaluthara

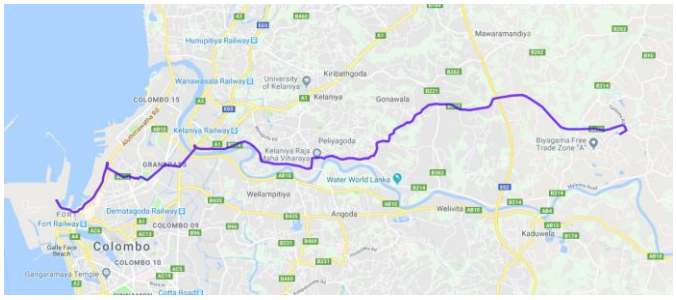


Figure 5.3 – Route followed from Colombo Port to Biyagama industrial Zone

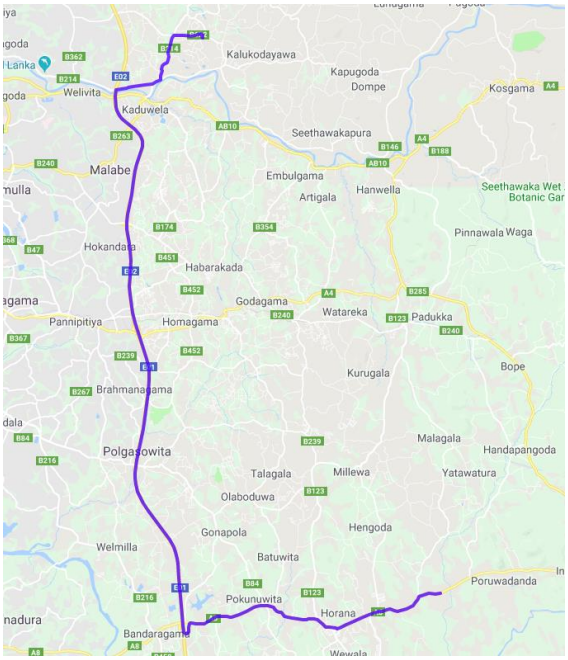


Figure 5.4 – Route followed from Biyagama Indurial Zone to Ingiriya Bodyline

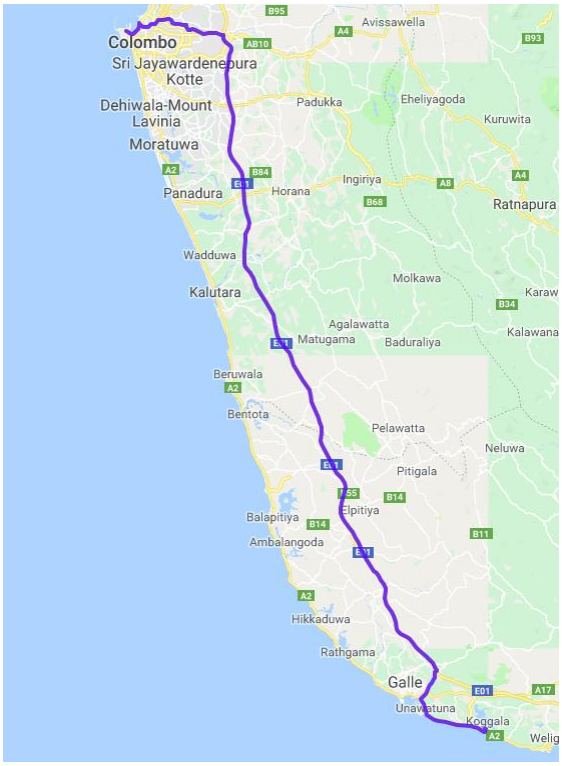


Figure 5.5 – Route followed from Colombo Port to Koggal MAS induies

5.2 Approach 1 – Evaluation Results

Framework is executed with simulated IoT data which is passed to the framework via a web user interface. As the result of successful contract deployment the contract address received. This result indicate that the indented smart contract consist of all the state variable to perform future transactions that are associated with it. The successful deployment of the smart contract made suppliers account balance deducted by 50ETH as which is the value of the specified smart contract shown in the figure 5.6.

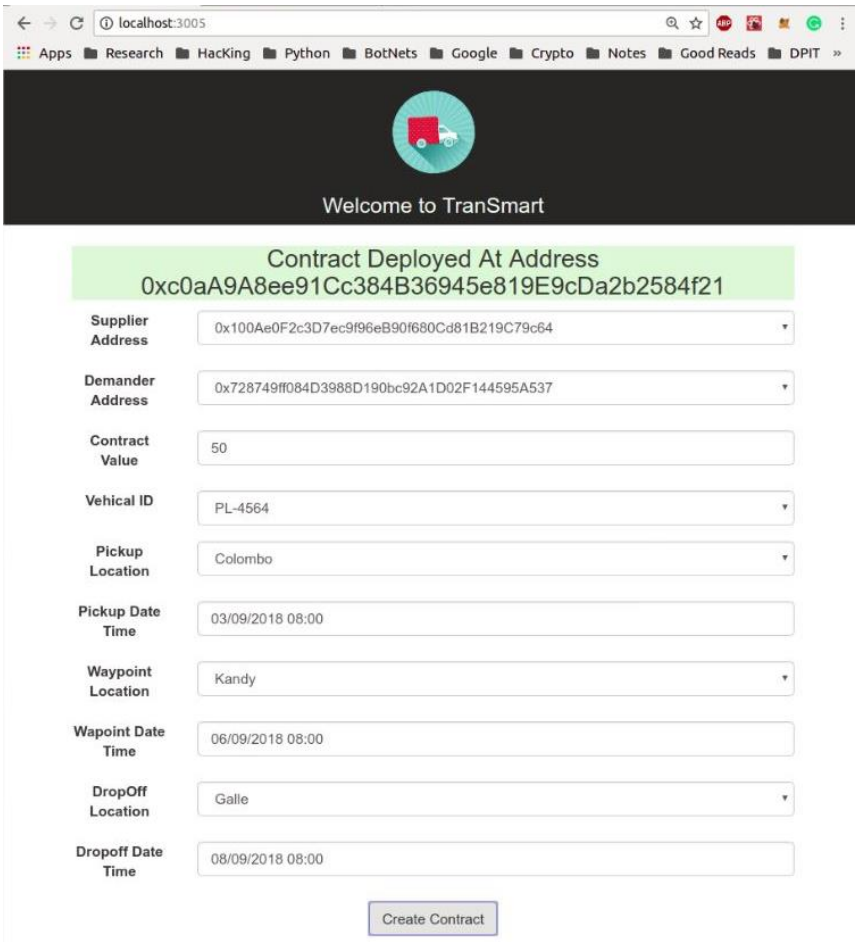


Figure 5.6 – Contract Creation UI

Figure 5.7 depicts the results from successful Smart contract deployment into the smart blockchain. This contract creation event has consumed 7555735 Wei and it is deployed o the blockchain address 0xc0aA9A8ee91Cc384B36945e819E9cDa2b2. Figure 5.8 shows how the account balances haven been change. The demander's account is deducted by 50ETH this amount is escrowed to Smart contract as its value.

Figure 5.7 – Contract creation viewed from Ganache UI

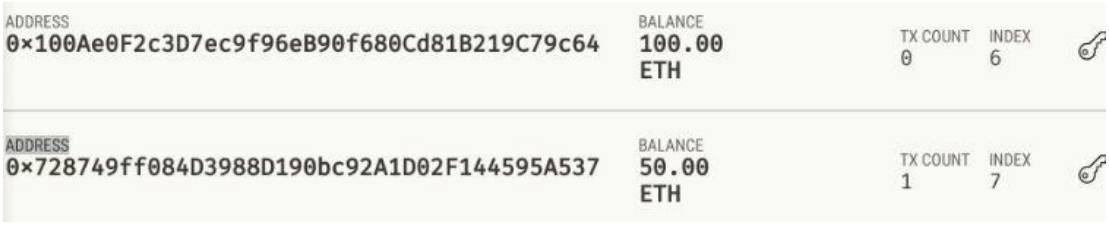
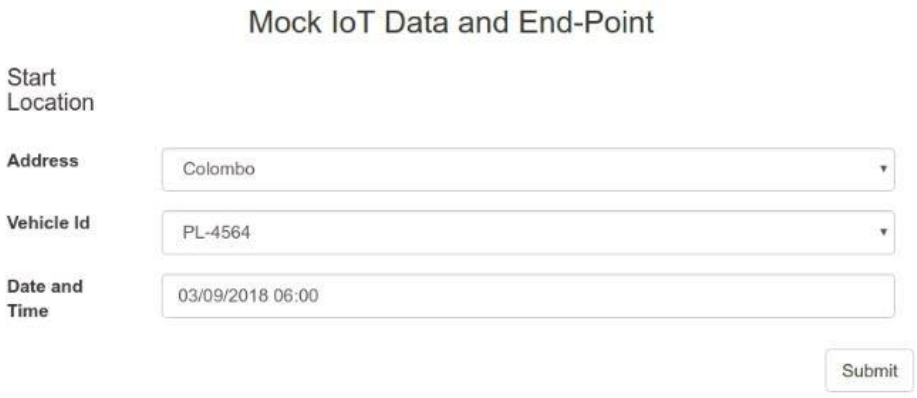


Figure 5.8 – Change in account balances after contract creation

To simulate IoT data web interface is used where the values of IoT data oracles are given as user inputs to the framework. In this simulation server time is not used as the time of the data because then the simulator will have wait for longer time periods to maintain time intervals in between data feeds.



When IoT data from data oracles are passed to the framework the framework executes as expected and produced final results where the supplier's account is added with 50ETH that is previously escrowed to the smart contract (Figure 5.9). This procedure is done with 10 Ethereum accounts over 50 times and the results were 100% similar to the expected results.

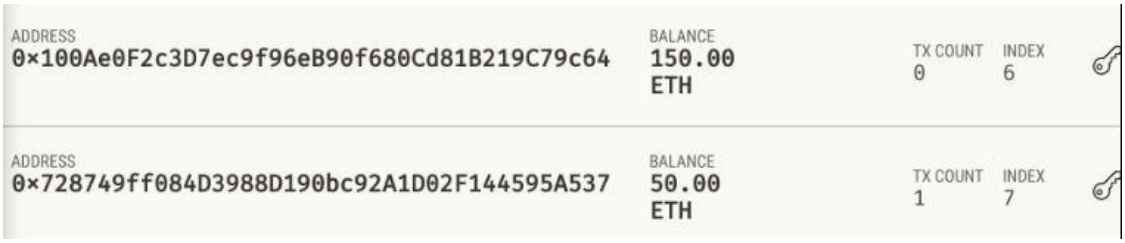


Figure 5.9 – Final account balances after transaction execution

5.4 Approach 2 – Evaluation Results

Quantitative analysis carried out to find out mean and the variance value of Gas amount spend on contract creation and contract method calls. Contract creation is done when two parties agreed to contract and deployed on the blockchain. Contract method calls occur when data oracles send data to the framework. For contract creation ten (10) smart contracts were created and data was collected via Ganache GUI. Data from the contract method was collected by performing method calls to contracts ten (10) times.

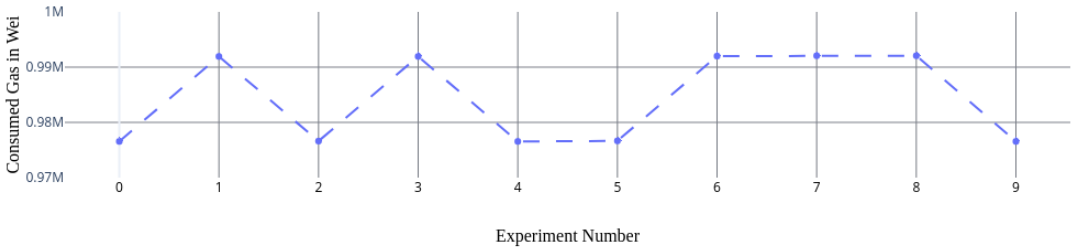


Figure 5.10 – Gas values consumed for contract creation

	Value in Wei
Mean	984291.40
Variance	66010675.37
Sample standard Deviation	8124.69

Mean Gas consumption for contract creation is 984291.40 with a sample standard deviation of 8124.69 in Wei. Therefore Gas consumption for contract creation can be assumed as a constant value when it is compared with Ether. This assumption is supported by the fact the contract values are in Ether range and standard deviation in Gas consumption for contract creation is 8124.69 x 10^{-15} in Ether. (1 Wei = 10^{-18} Ether). Results from this evaluation step interprets that Gas consumption for smart contract creation is insignificantly small.

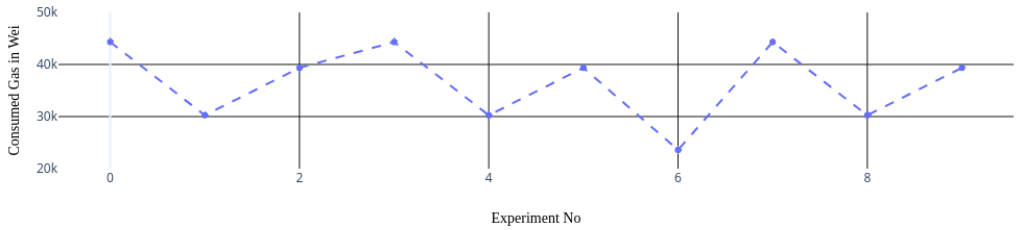


Figure 5.11 – Gas values consumed for contract method calls

	Value in Wei
Mean	36536.00
Variance	54546444.66
Sample standard Deviation	7385.55

Mean Gas consumption for contract method calls is 36536.00 with a sample standard deviation 7385.55. This value is insignificantly small when compared to the total value of the Smart Contract. Contract method calls are initiated from the off-chain computation layer in the framework when state change is identified. Design method followed in framework to perform state transitions deification as an off- chain computation largely supports to minimize the number of method calls that is required until Smart contract auto liquidation. Number of total method call can be calculated as follows for a transport job.

$$T_M = N_L (L_C + B_C) + T_C$$

Where,

T_M = Total Number of Method Calls + Temperature method call

N_L = Number of Locations

L_C = No of Location validation method calls to identify state change

B_C = No of Barcode validation method calls to identify state change

T_C = No of Temperature validation method calls to identify state change

By using the above formula total gas consumption until Smart Contract auto-liquidation can be calculated as follows for this framework

$$T_G = T_M * G_M + G_C$$

Where,

T_G = Total Gas Consumption

T_M = Total Number of method calls

G_M = Gas consumption per method call

G_C = Gas consumption for Contract creation

Proposed smart contract uses three locations. Smart contract receives only one method call when a state change is occurred (This is because off-chain computation layer identifies state change and perform method calls only when state change occurs) Total average gas consumption for the proposed Smart contract is calculated as follows. By applying above values to first equation resulted value is seven.

Appling seven as total number of method calls in the second equation along with the mean Gas consumption values obtained above produces the result as average total Gas consumption is $1.2400434 \times 10^{-11}$ Ether. This is insignificantly small value when compared to the Smart contract value.

5.4 Summary

This chapter presented the obtained results from the proposed framework in different stages of execution of the framework. Evaluation of the framework is carried to practical applicability and Gas consumption of the framework. The results indicate that proposed framework is feasible for real world deployment and use of smart contract is feasible solution to establish trust. Results of this framework indicate that proposed framework is capable to serve as foundation for a larger platform which uses blockchain and driven by smart contracts.

Chapter 6

Conclusion

6.1 Introduction

This chapter reviews the research aim, objective and research problem with obtained results. Future works and limitation of this study is stated in the later part of this chapter. Finally this chapter includes practical application is stated in later part.

6.2 Conclusions on Research Aim and Objectives

The aims in this research is to create a framework for transport and logistics industry which is based on blockchain and driven by smart contracts. With use of smart contracts agreement between demanding and suppling parties was formulated with three condition variables (Location and Date Time, Bar Code, Temperature). As results indicate the framework is able to create smart contract when the values are given as input. The framework consist of web interface to make this task easier.

The study had objective to use multiple smart contract to formulate transport contract agreement. With design proposed for Smart contracts and with the use of off-chain data store it is identified that multiple smart contract are not necessary to formulate transport contract agreement. The proposed Smart contract consists of one auto-liquidation state is. The auto liquidation state is that vehicle arrival to the final drop-off location. The necessity for multiple liquidation points is avoided designing the Smart contract that uses state variables to reflect its current state in a given moment. Using state variable values at the end of the transport job liquidation is defined. The proposed smart contract design has performed as expected fulfilling requirements in the transport and logistics industry under the scope of the research.

The transport contract agreement is violated when at least one type of conditions are not met by the supplying party. Penalties for each type of condition violation is agreed upon creation of the Smart contract. By using this approach when a contract violation occurs it is resolved in a way neither the supplier nor demander will gain profit. The violated smart contracts sends back the penalty amount to demanding party to zero out unexpected profit which is otherwise gained by supplier.

Initial Coin Offering is designed for the completeness of the framework. By using its own token for transactions the framework maintains its integrity. Initial Coin offering proposed via performing an ERC20 token sale. Using Smart contracts. For this purpose, a new token was created and used in the framework. Using this token new user can onboard to the framework.

6.3 Conclusions on Research Problem

Transport job consists of multiple types of conditions or terms. Among these conditions, cargo delivery times and locations are one important in industry level transportation. The delivery time for each location is captured in the contract creation stage and the location and time values are implemented in Smart contract using two state variables. The next important condition is that contents of the cargo. This condition reflects the fact that transportation is done for correct cargo. This is captured in the contract creation stage using barcode value and implemented in Smart contract using another state variable. The third variable that was the temperature which is important for sensitive cargo. This condition was integrated into the smart contract using a state variable. Final proposed smart contract design was able to perform well with data using these state variables and identified contract condition violations using them.

Data oracles provide data to smart contracts to perform state transitions. Data oracles send raw live data to the framework. Therefore proposed design process them before sending to Smart contracts. This off-chain processing is done to identify state transitions. Using this largely reduced on-chain computations which would otherwise cost Gas from the Smart Contract. Important data that indicates state transition was passed to smart contract using ABI and Smart contract address. This indirect

connection of data oracles to Smart contract performed well since this approach only costs an insignificantly small amount of Gas until auto-liquidation step as stated in the second approach of the evaluation chapter.

6.4 Limitations of the Research

Proposed implementation for the Smart contracts consists three locations per a transport job. To accommodate a higher number of locations Smart contract should be rewritten and integrated into the framework. This limits the number of scenarios that this framework can be directly applied. The Framework supports for only two stakeholders that hold a transport agreement between them. The design of the Smart contract should be changed when the higher number of conditions, multiple stake holders are essential in the transport agreement. Smart contract design only allows single payment scheme which is 100% payment after final delivery.

The Gas consumption for Smart contract execution is a significant factor when the number of contracts increases by a particular account. This effect is unavoidable since Gas consumption is compulsory for Smart contract execution and maintain the validity of the transactions.

Proposed framework directly rely on the data from data oracles for transaction execution. Performance and validity of data oracles affect the transaction execution. Data oracles send data via internet protocols to the framework. Network failures or corrupted data might result in incorrect transactions or Smart contract which will be never executed (due to data loss).

Proposed ICO is vulnerable to short address attacks. Which is equivalent of minor SQL injection bug. EVM appends 0at the end of an address if it detects an underflow (a smaller address that doesn't consist 256 bits) which happens when dealing with data types that can be up to 256 bits, EVM takes a sane approach to appends 0.

6.5 Implications for Further Research

The Smart contract design in this framework can be improved to accommodate multiple stakeholders, multiple conditions and payment schemes that perform transactions upon partial fulfilment of transport job. Smart contract creation procedure can be improved to include the separate smart contract creation for each type of contract violation. To improve the validity of data oracles each data oracle can be assigned with account address with zero value which will include data oracles into the network.

The legal state of ICO is mostly undefined. Ideally, the token is sold not as a financial asset but as a digital good. In this case, in the most jurisdiction, the funding with an ICO is not regulated, which makes it extremely easy and paperless, given a lawyer experienced with the issue is on board. Therefore a better solution for ICO will expand this framework.

Using a public blockchain instead of a private blockchain will make this framework open to worldwide transport and logistics platform. Final expansion of this proposed framework can be decentralized autonomous organization that performs in a global level.

References

[1]: Logistics Market - Global Industry Analysis, Size, Share, Growth, Trends, and Forecast 2016 to 2024 - http://www.transparencymarketresearch.com/logistics-market.html [Accessed 10th July 2018]

[2]: Nakamoto, Satoshi. Bitcoin: A Peer-to-Peer Electronic Cash System, 2009

[3]: Buterin, Vitalik. A Next-Generation Smart Contract and Decentralized Application Platform

[4]: Luu, Loi and Chu, Duc-Hiep and Olickel, Hrishi and Saxena, Prateek and Hobor, Aquinas. Making Smart Contracts Smarter. 2016 254-269. 10.1145/2976749.2978309.

[5]: Wood, Gavin. Ethereum: a secure decentralized generalized transaction ledger. 2018 May 04

[6]: Nach, Hamid. Blockchain and smart contracts in the logistic and transportation industry

[7]: Hribernik, Karl. A An internet of things for transport logistics - an approach to connecting the information and material flows in autonomous cooperating logistics processes

[8]: Rajbhandari, Rajat. Exploring Blockchain: Technology behind Bitcoin and Implications for Transforming Transportation

[9]: Christidis, Konstantinos and Devetsikiotis, Michael. Blockchains and Smart Contracts for the Internet of Things. IEEE Access. 4. 1-1. 10.1109/ACCESS.2016.2566339. (2016)

[10]: Badzar, Amina. Blockchain for securing sustainable transport contracts and supply chain transparency

[11]: Smart Contracts: 12 Use Cases for Business & Beyond, Chamber of Digital Commerce Washington, D.C. 2016 December

[12]: Quasa Framework : https://www.quasa.io [Accessed 17th June 2018]

[13]: Szabo, Nick. Formalizing and Securing Relationships on Public Networks, 1997 First Monday. 2. 10.5210/fm.v2i9.548.

[14]: Online ship tracking system of Maersk shipping line.

https://www.maerskline.com/en/routes/track-shipments [Accessed 05th August 2018]

[15]: Container tracking Platform: https://www.track-trace.com/container [Accessed 11th April 2018]

[16] IBM Maersk Shipping line collaboration [Accessed 22th March 2018]

https://www.ibm.com/blogs/blockchain/2018/01/digitizing-global-trade-maersk-ibm/

[18]: https://www.cognizant.com/whitepapers/blockchains-smart-contracts-driving-the-next-wave-of-innovation-across-manufacturing-value-chains-codex2113.pdf

[19]: Álvarez-Díaz, Néstor & Herrera-Joancomartí, Jordi & Caballero-Gil, Pino. (2017). Smart contracts based on blockchain for logistics management. 1-8. 10.1145/3109761.3158384.

[20]: Blockchain In Logistics: DHL Report 2018 [Available at https://www.logistics.dhl/content/dam/dhl/global/core/documents/pdf/glo-core-blockchain-trend-report.pdf]

[21]: Yao-Chieh Hu, Ting-Ting Lee, Dimitris Chatzopoulos Hierarchical interactions between Ethereum smart contracts across Testnets. In CryBlock'18 Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems

[22]: Xu, X., Pautasso, C., Zhu, L., Gramoli, V., Ponomarev, A., Tran, A.B. and Chen, S., 2016, April. The blockchain as a software connector. In Software Architecture (WICSA), 2016 13th Working IEEE/IFIP Conference

[23]: ZAP data oracle framework: https://www.zap.org/whitepaper.pdf [Accessed 10th August 2018]

[24]: Oracalize Framework: http://docs.oraclize.it/#home [Accessed 10th August 2018]

[25]: Mahdi H. Miraz, Maaruf Ali, Applications of Blockchain Technology beyond Cryptocurrency. In Annals of Emerging Technologies in Computing (AETiC), Print ISSN: 2516-0281

[26]: Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels Town Crier An authenticated Data feed for Smart Contracts. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security

[28]: TLSnotary – a mechanism for independently audited https sessions. [Accessed 20th August 2018] https://tlsnotary.org/TLSNotary.pdf

[29]: Mengru Tu, (2018) "An exploratory study of Internet of Things (IoT) adoption intention in logistics and supply chain management: A mixed research approach", The International Journal of Logistics Management, Vol. 29 Issue: 1, pp.131-151

[30]: Chang, S.I., Hung, S. Y., Yen, D. C. and Chen, Y. J. (2008), “The determinants of RFID adoption in the logistics industry - a supply chain management perspective”, Communications of the association for information systems, Vol. 23 No. 1, pp.12.