Letter-of-Credit Chain: Cross-Border Exchange based on Blockchain and Smart Contracts

Khoi Le Quoc[1], Phuc Nguyen Trong[2], Hieu Le Van[3], Hong Khanh Vo[4], Luong Hoang Huong[5], Khoa Tran Dang[6], Khiem Huynh Gia[7], Loc Van Cao Phu[8], Duy Nguyen Truong Quoc[9], Nguyen Huyen Tran[10], Huynh Trong Nghia[11], Bang Le Khanh[12], Kiet Le Tuan[13]
FPT University, Can Tho City, Viet Nam

Abstract— **The exchange of goods between countries is growing, contributing to the promotion of logistics-related technologies. More and more systems are adopting advances in science and engineering to reduce manual handling steps, thereby reducing transit time. Letter-of-Credit (LOC) is a standard method where the parties involved will enter into agreements for the sale and exchange of goods. Specifically, each party will receive a set of original documents and does not need to meet face-to-face under the bank's witness. The process brings many benefits in terms of time and reduces records processing. However, the system faces a lot of risks when one of the parties is dishonest. On the other hand, the traditional LOC systems face a lot of risks related to the transparency of information about the goods, and also the supplier may lose the goods (e.g., 4/100 Vietnamese cashew nut containers are lost. stuck in Italy) or deposits in the hands of shipping companies (e.g., GNN Express - Vietnam) and many more. To this end, many research directions have exploited blockchain technology and smart contracts. Specifically, all information related to the transaction between the supplier and the demander including package, time, and delivery location. However, there needs to be a mechanism to ensure the smooth implementation of smart contracts, specifically for sanctioning when there is a conflict between a supplier and a demander. This role should be considered for the transaction manager, who directly designs and is responsible for their smart contracts. Currently, there is no mechanism to guarantee all interests of the parties involved in non-bank transactions. To increase the processing capacity and integrate with the Blockchain system, we propose the Letter-of-credit Chain that defines the agreements between the parties in international trade. We also deploy the proof-of-concept of the Letter-of-credit Chain on the three EVM-supported platforms (i.e., under ERC20), namely, Ethereum, Binance Smart Chain, and Fantom. By evaluating the actual execution of Gas for each platform, we found that our proposed model had the cheapest fee when deployed on the Fantom platform. Finally, we share the deployment/implementation of these platforms' proof-of-concept to encourage further future research.**

Keywords*—*Letter-of-Credit; blockchain; smart contract; authorization; Ethereum; Fantom; Binance smart chain

I. Introduction

It is undeniable that the development of technology has changed almost completely the approach to a business organization as well as a management strategy to meet customer requirements [1]. The exchange of goods no longer takes place in a narrow area limited in area, but instead, companies can transport and export/import goods from all over the world. Traditional international trade models were originally built on trust. Specifically, in the first stage, the supplier will bring the goods to another city or country to find a demander. It is clear that this model is very risky for both the supplier and the demander. In particular, demanders may purchase inferior products because all constraints are not controlled; on the supplier's part, they may lose the entire goods if the demander refuses to pay or the product is past its expiry date due to the time-consuming shipping process [2]. In order to solve the risks of transporting the goods, the supplier will authorize the transport company when transporting the goods to the demander (i.e., minimizing risks in transportation and transit time), i.e. that the parties can communicate indirectly through intermediaries instead of having to meet face-to-face - this also reduces the costs incurred. Due to the growing and expanding trade, where the need to transport goods increases and the transit time decreases. Therefore, both parties will authorize a trusted third party called a transaction manager (e.g. a bank). All contractual requirements and constraints must be accepted by all three parties (i.e. supplier, demander and third party). The role of the intermediary is assigned to the Bank (called Letter-of-Credit - LOC). Specifically, the demander is provided with an economic guarantee from the bank that grants credit to the exporter of the goods [3]. The supplier receives the money only if and only when the demander receives the goods and provides all statements related to the shipment confirmed by the transaction manager. An important disadvantage of the traditional LOC model is that it is easy for suppliers to lose goods if they work with the untrusted transaction manager and malicious demanders. One of the most examples of this problem is introduced in Section IV, which present the cashew nut export from Vietnam to Italy in 2022[1].

E-commerce has enhanced the process of transporting goods across borders thanks to the exchange of routes through e-commerce platforms. This process is made faster and brings many benefits to both the supplier and the demander. A series of e-commerce platforms (e.g., Amazon, Alibaba) have largely changed users' shopping habits, while shipping companies (e.g., FedEx, ASL) have also accelerated the conversion process. goods. The freight conversion process is based on a Cash-on-Delivery (COD) shipping company, where the carrier will play an extremely important role in delivering and receiving the demander's funds. Payment to the supplier is the responsibility of the carrier. Most shipping companies will keep the demander's money before handing it over to the supplier. If the shipping company goes bankrupt or refuses to pay, the supplier will lose money [4]. In fact, a series of shipping companies have appropriated the supplier's money (e.g., GNN Express).

[1]https://english.vov.vn/en/economy/vietnam-requests-italy-to-investigate-suspected\ \-cashew-nut-export-scam-post931226.vov

To solve these challenges, several Blockchain-based approaches have been developed to replace the traditional LOC model. These new systems provide a secure platform for both suppliers and demanders, aka decentralized marketplaces/exchanges [5]. These protocols focus on dealing with issues related to suppliers, demanders, and carriers (see details in the III section). However, these models still do not meet the requirements of international trade, where the role of the transaction manager is extremely important [6]. To solve this problem, we introduce Letter-of-Credit Chain, a system based on Blockchain and smart contracts to solve the insurance problem for cross-border exchange. Letter-of-Credit Chain builds three main user groups, including suppliers, demanders and transaction manager. This model consists of eight main steps, from the supplier creating the package/goods to the order being delivered to the demander and the order arrival. Besides, we develop a role-based access control model (i.e., authorization) to define logical constraints in smart contracts to maintain the stable operation of the system. To define the logical binding between stakeholders on smart contracts, we also exploit the Solidity. To evaluate the Chain of Credit, we deployed a test model for our proof-of-concept on all three of the most popular platforms that currently support the Ethereum Virtual Machine (EVM) environments, including Ethereum, BNB Smart Chain, and Fantom (see [7] for further info). To support the current letter-of-credit system (i.e., International Trade), we share our proof-of-concept implemented on all three platforms.

Following this section, we present the background of blockchain technology and its platform in Section II. Whereas, Section III describes the summary, limitations and challenges of the current approaches. Then, the two next sections define the problem statement of the traditional model of the Letter-of-Credit approach and also introduce the architecture of Letter-of-Credit Chain Architecture based on the blockchain and smart contract in Sections IV and V, respectively. Section VI describes the proof-of-concept of the Letter-of-Credit Chain, i.e., data structure, execution algorithm, and authorization. Last but not least, Section VII focuses on the effectiveness proof via the evaluation process based on deploying Letter-of-Credit Chain in the ETH, BNB, and Fantom platforms. Finally, we continue with the conclusion and the future work of the article in the last section.

II. Background

A. Blockchain Technology

Blockchain was popular after the introduction of Bitcoin by Nakamoto in 2008 [8] and is usually represented as a transparent, trusted, and decentralized ledger. The blockchain-based system manages transaction data on multiple computers simultaneously on a peer-to-peer network. Therefore, it creates a secure connection between the transacting parties (i.e., receiver and money transmitter) without the need for a traditional third party (e.g., a bank) [9].

The most popular types of blockchains today include Public, Private, and Hybrid (Called Consortium). In the first type, the two best-known examples of public blockchains currently are Bitcoin and Ethereum. Any users (including hidden ones) could join to Blockchain network to view content, execute a new transaction, or check the integrity of the existing blocks. For private blockchains, some common examples of this type include GemOS, MultiChain, Ripple, and Eris. Unlike the Public type, they only support authorized users who can join the network as well as execute, check transactions to the block or create a new block [10]. Combining the two, a semi-private (called Consortium) blockchain is defined as the boundary between public and private ones. It strives to achieve outstanding characteristics in each category - specifically, Consortium blockchains are often deployed for enterprises to ensure their security and interact with their partners for better business. Two famous examples of hybrid blockchains are Hyperledger Fabric [11] and Ethereum [12] (i.e., which allows the creation of Golang-based federated blockchains).

B. Blockchain Platform

1) Ethereum: Ethereum [13] is a decentralized blockchain platform for running smart contracts with the support of the Solidity programming language. Similar to other high-level languages (e.g., Java), Ethereum is executed by the Ethereum Virtual Machine (EVM). Ethereum supports decentralized finance (DeFi) protocols where smart contract-based constraints are provided.

2) Hyperledger Fabric: Hyperledger Fabric [11] is open-source enterprise-grade permission designed for large-scale commerce. This platform is designed based on distributed hyper-ledger mechanism and supports both public and private blockchain platforms simultaneously. Hyperledger Fabric and Ethereum together perfect Turing. However, instead of executing smart contracts on the EVM virtual machine like Ethereum, Hyperledger code is executed in Docker containers called ChainCode. It allows developer applications to deploy smart contracts with minimal overhead. Another advantage over Ethereum is that it supports high-level programming languages (i.e., Java and Go) instead of relying on Solidity. With this advantage, Fabric has facilitated the development and maintenance of the platform. By not having to switch to a new language, Fabric has helped to reduce operating costs (e.g., system maintenance, information storage and querying within the blockchain).

C. Smart Contracts

A smart contract (Ethereum) or chaincode (Hyperledger Fabric) is a term that describes a set of protocols that assist developers in defining terms and agreements in transactions between the parties to the contract. The entire process of smart contracts is not dependent on external interference and is performed automatically based on the support of Blockchain technology. In a Dapp, the terms and constraints of a smart contract are recorded in the language of a computer and are equivalent to a legal contract.

1) Characteristics: The Smart Contract routine has the following characteristics:

- **Distributed**: Replicated and distributed in all nodes of the Ethereum network. This is one difference from other solutions based on centralized servers.
- **Deterministic**: Only take actions that they are designed to perform if the conditions are satisfied. Be-

sides, the results of Smart Contracts remain the same no matter who the executor is.

- **Automate**: Able to automate all kinds of tasks, and it works like a self-executing program. However, in most cases, if the Smart Contract is not activated, it will remain "inactive" and will not perform any action.
- **Non-modifiable**: Smart Contract cannot be modified after deployment. They can only be "deleted" if this function has been added before. Therefore, it can be said that Smart Contract is like an anti-forgery code.
- **Customizable**: Before deployment, Smart Contracts can be encoded in different ways. So, they can be used to create many types of decentralized applications (Dapps). Ethereum is a blockchain that can be used to solve any computational problem (Turing complete).
- **No need to rely on trust**: Two or more parties in a contract can interact through a Smart Contract without knowing or trusting each other. In addition, blockchain technology ensures the accuracy of data.
- **Transparency**: Since Smart Contracts are based on a public blockchain, no one can change their source code, although anyone can view it.

2) How Smart Contracts Work: The working principle of a smart contract can be compared to a vending machine. It only automatically executes pre-programmed commands.

First, assets and contract terms are both encrypted and transferred into a block on the Blockchain. Then this smart contract will be distributed and copied by the nodes working on that platform. After receiving the deployment order, the contract will be deployed according to the predetermined terms. Simultaneously, the smart contract will also automatically check the implementation of the commitments stated in the agreement.

- Cost savings: Pay a minimal fee to the blockchain network, saving fees.
- Flexibility: The terms in the contract are handled flexibly and efficiently for the user.
- Transparency, clarity: all payment transactions can be traced, but payment transactions will not be reversed at all, and all transactions will be recorded on the blockchain with extreme clarity.
- High Reliability: Once the contract is completed, no one or a party can interfere in the execution and negotiation of the contract.
- Fast, convenient: can set up and execute a contract in seconds, install for many people simultaneously, and use it many times.

3) Solidity: Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs that oversee the conduct of records inside the Ethereum state. Solidity was influenced by JavaScript, C++, and Python and is intended to focus on the Ethereum Virtual Machine (EVM).

Solidity is statically composed and upholds inheritance, libraries, and complex client-characterized types, among different highlights. With Solidity, you can make contracts for utilizations like democratic, crowdfunding, dazzle barters, and multi-signature wallets.

4) Web3.js: Main steps in creating blockchain applications with Ethereum:

- Innovative contract development - composing code that gets sent to the blockchain with the Solidity programming language.
- It is creating sites or customers that cooperate with the blockchain - composing code that peruses and contains information from the blockchain with smart contracts.

Web3.js empowers you to satisfy the following duty: creating customers that communicate with The Ethereum Blockchain. An assortment of libraries permits you to perform activities like sending Ether starting with one record and then onto the next, peruse and compose information from shrewd agreements, make smart contracts, and thus significantly more.

If you have a web advancement foundation, you may have utilized jQuery to settle on Ajax decisions to a web worker. That is a decent beginning stage for understanding the capacity of Web3.js. Rather than using jQuery to peruse and compose information from a web worker, you can utilize Web3.js to peruse and keep in touch with The Ethereum Blockchain.

Web3.js converses with The Ethereum Blockchain with JSON RPC, which means ”Remote Procedure Call” convention. Ethereum is a distributed organization of hubs that stores a duplicate of all the information and code on the blockchain. Web3.js permits us to make solicitations to an individual Ethereum hub with JSON RPC to peruse and compose information for the organization. It's similar to utilizing jQuery with a JSON API to peruse and manage data with a web worker.

5) Remix: The remix is a Solidity IDE used for writing, compiling, and debugging Solidity code. Solidity is a high-level, contract-orientated programming language for writing clever contracts. It is affected by popular languages such as C++, Python, and JavaScript.

Ethereum is a general-purpose blockchain, which is more suitable for using advanced scripts (also known as smart contracts) to describe business logic. Ethereum is developing as a decentralized or global computer that combines blockchain functions with a broader perspective. As a reliable machine with a complete Turing contract engine.

Benefits of using remixed IDE to compile and deploy smart contracts:

- Compile the contract in Remix IDE.
- See a few warnings issued with the help of using a compiler while high-quality practice is not followed.
- Contract implementation on JavaScript EVM (Ethereum Virtual Machine).
- Make transactions with the implemented contracts.
- See example reading and writing in the IDE terminal.

D. Our Selection Platform

As mentioned in the Introduction section, this article deploys the proof-of-concept of the Letter-of-credit Chain on three platforms, namely, Ethereum, Binance Smart Chain, and Fantom[2].

1) Binance Smart Chain: Binance Smart Chain (BSC) is an enhanced version of the original Binance Chain version. BSC is designed to be a parallel platform to the first version. Similar to Ethereum, BSC offers Dapp developers options to support smart contracts and can be deployed on other Blockchain platforms that support EVM.

2) How does Binance Smart Chain Work?: BSC applies a hybrid model of Proof of Authority and Proof of Stake - called Proof of Staked Authority (PoSA). Validators for the BNB system will put a certain amount of BNB into the system and receive a bonus after each successful validation.

Binance Chain and Binance Smart Chain are cross-chain compatible and designed to be completely in sync. With BSC, assets can be moved between blocks thanks to the fast transaction capabilities of the original version and smart contracts of the improved version (i.e., EVM integration). Specifically, Binance Chain supports two tokens (BEP-2 and BEP-8). In addition, Binance Chain can also be swapped with Smart Chain BEP-20 tickets. So, thanks to ERC-20 contract compatibility, DApp developers on other EVM-enabled platforms can switch to Binance Smart Chain relatively quickly.

III. Related Work

Several protocols exploited the blockchain system's advantage to improve their transaction among the peers or components. Some of them considered Cash-on-Delivery (COD) model [14], [15] (i.e., suppliers, demanders), the medical care system [16], [17], [18] (i.e., doctors and patients), health care emergency situation [19], [20] (i.e., medical staff and the patients' friends or relatives), or blood donation - humanitarian blood transfusion (donors and recipients) [21], [22] and much more.

To prove the improvement of Blockchain for traditional shipping (i.e., Letter-of-Credit shipping (called LOC) or COD) Ha et al. [6] described the current shipping system faced the massive drawback (e.g., dependence on trusted third parties, goods/order management, complicated payment processes among the parties in the same peer or ecosystem, losing the package and deposit for supplier and demander, respectively. To this end, Le et al. [23] suggested that blockchain technology could fill these gaps via the usage of smart contracts and decentralized management in COD general and LOC special. For instance, the Ethereum ecosystem proposed a method called `localEthereum` which is introduced to support the transaction or DeFi Dapp between the suppliers and demanders [24]. Similarly, `OpenBazaar` [25] developed based on the `localEthereum` extension, in which this protocol defined the demander and supplier-sponsored. However, compared to `localEthereum`, the main difference between `OpenBazaar` was that the `OpenBazaar` involved the three parties: the supplier, the demander, and the moderator (i.e., a new role in control).

Moreover, a new protocol targeted at helping transport products from suppliers to demanders [26] exploited the ETH-based transaction to propose a COD/LOC mechanism. In particular, this tool considers the new actor (i.e., shipper) rather than focusing on only the transaction between the demander and supplier as in the previous approaches above. However, this approach still has the main drawback is that it required trusted behavior from the shipper (i.e., new role) not only in this task but also the interaction between the supplier and demander (i.e., how can be proof of this level is still the open question). Furthermore, there are impossible to assume that the stakeholders are trusted for all their behavior in the system/network. Hence, this is the main limitation of these approaches.

To increase the shipper's role in the blockchain-based approaches, some studies (e.g., [27] and [28]) introduced carriers in the decentralized marketplace, rather than focusing only on demanders and suppliers. These works developed the transportation processes as well as provided the mechanisms to promote and ensure the benefit of the stakeholders. These models also penalized any parties who intentionally commit fraud; therefore, the demanders' and suppliers' interests are enhanced. However, the scope of these models' application has limited to the distance of shipping where the transaction occurs in the same city or at most in the same country. Last but not least, they have not considered the conflict issues among the parties in the same transaction, for instance, suppliers and demanders.

Compared to the existing works, our proposed model (i.e., Letter-of-Credit Chain) introduces a trusted international trade channel that connects all the suppliers and demanders around the World (among the different countries). We aim to reduce the conflict among the parties in the system to solve the two case studies, namely, GNN Express as well as cashew nut export problems as present in the Introduction section.

IV. Problem Statement

Fig. 1 describes the problem of the cashew nut export from Vietnam to Italy in 2022.

The Vietnamese enterprise (Exporter) first signs a contract with the Italian importer to specify the contract's bank and port of destination. The Exporters then deliver goods to shipping lines and receive "original documents". Apter that, the Exporter receive the original documents and bring them to a bank in Vietnam (a.k.a Exporter's bank) to ask this bank to collect money. The Exporter's bank next send this set of original documents to the importer's bank in Italy. At this point, the Italian importer's bank will pay the exporter's bank and give this original document to the Italian importer for them to receive the goods.

However, the problem with this is that the importer's bank in Italy did not receive the original set of documents, but they get photocopies instead. Thus, this bank refuses to transfer money.

[2]Since selected platforms supporting the EVM are similar in the execution process (BNB Smart Chain and Fantom), this section gives summarize of the Binance Smart Chain in a nutshell.

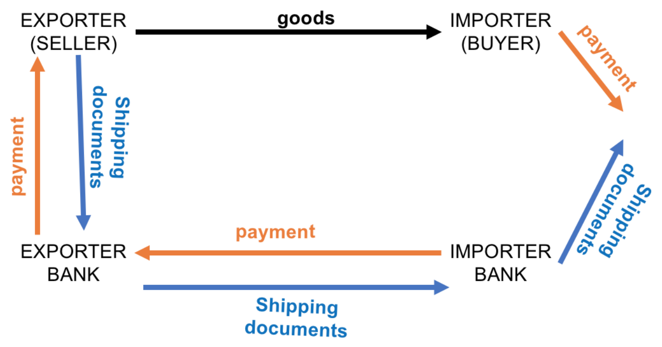

Fig. 1. The Problem Statement of the Cashew Nut from Vietnam to Italy in 2022.

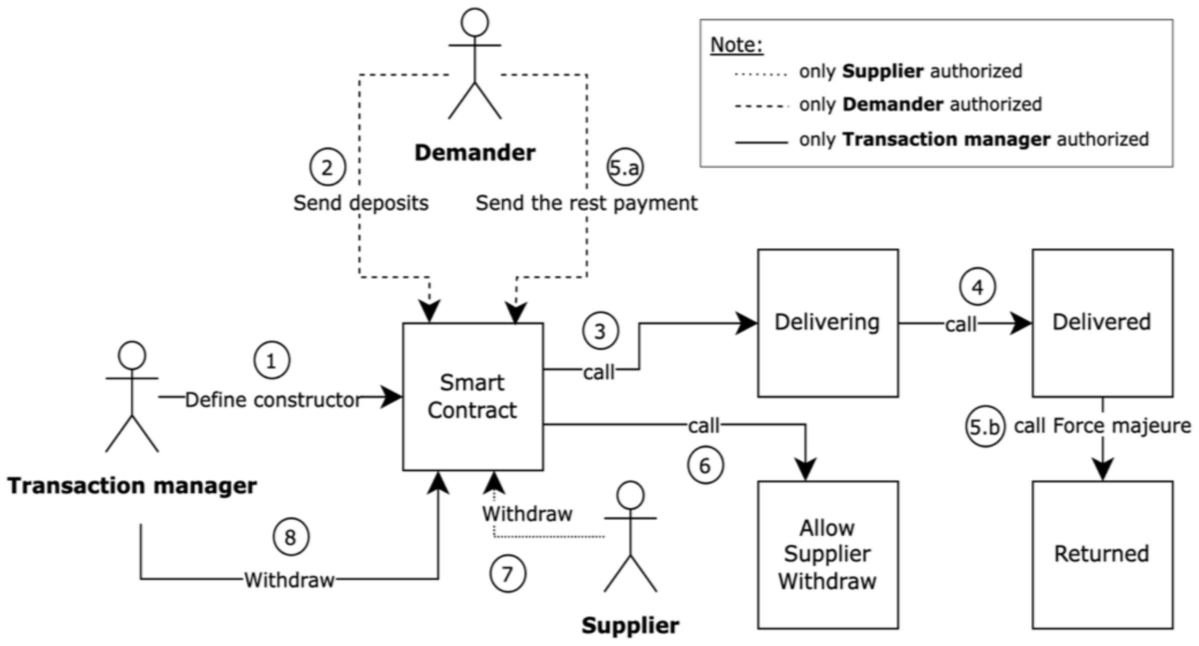

Fig. 2. Letter-of-Credit Chain Architecture.

V. Letter-of-Credit Chain Architecture

Fig. 2[3] shows Letter-of-Credit Chain architecture including three main components, namely supplier, demander, and transaction manager.

- *supplier*: The party who provides goods or services to another organization. In Letter-of-Credit Chain architecture, the Supplier can withdraw the money after the goods are delivered to Demander via the smart contract.
- *demander*: Who makes requests for an item or service in the platform economy. In this architecture, the Demander must send the amount of deposit as well as the remaining money to the Suppier via smart contract. In the case that Demander won't send the remaining amount, their deposit is lost. We also consider this point in our experiments (see Section VII).

[3]In this model architecture, we do not refer to off-chain tasks.

- *transaction manager or Trustee*: the third party who manages both traditional contracts and smart contracts. *TransactionManager* controls the two flows i.e., the order to Demander and the money to Supplier.

The TransactionManager define the smart contract (a.k.a. the tradition contract) and upload this on the Letter-of-Credit Chain [Step 1]. This can identity the address on the corresponding blockchain network. The second step show that the must be send the deposit to certain smart contract defined by the TransactionManager. In our approach rather than one-time-transfer the Demander can split this as multiple amount; however the sum of these amounts must be higher than 50% of the order's price. The order status is changed to Delivering in the third step, whenever the *TransactionManager* confirm that the shipping function is called. In the next step (i.e., the order delivered) we deploy this as off-chain. In the current version, we do not focus on the off-chain tasks in the whole transportation process. Step 4 calls the delivered function if and only if the order has arrived. We have two cases in this point, i.e., the Demander transfer the remaining amount in Step 5(a) or does not transfer in Step 5(b). Regarding to the behavior of the Demander we have two corresponding solutions that are i) if they transfer the remaining, the Supplier can withdraw their money for the goods/order via Allow Supplier Withdraw function in Step 6; otherwise the Demander lost their deposit and this amount automatically send to Supplier as shown in Step 7. Finally, the *TransactionManager* withdraw their amount based on the smart contract fee execution in Step 8 and does not depend on the Demander decision.

VI. Implementation

A. Data Structure

Fig. 3 describes the Letter-of-Credit Chain framework' data structure. We only consider the key information (i.e., Order or Goods) in this paper. The remaining ones are described in our code. Please follow the deployment of our implementation in the three platforms (see Section VII for more detail).

B. Algorithm

Letter-of-Credit Chain framework (i.e., Algorithm 1) executes from top to bottom. First, the TransactionManager add the smart contract code to the blockchain network. This point is created by the TransactionManager, but we can easily to detect the meaning of their requirement. In the next step we set up current_State; State.Created; and balance_Received = 0. When the Supplier uploads their orders to the network and finds the corresponding Demander (see lines 4 to 15) the Demander must sends deposit money to the smart contract. At this point, Algorithm 1 updates balance_Received as well as the smart contract also logs the history of the transaction into payment_Histories parameter. If the order on the delivering (current_State = State.Hold or State.Complete_Payment), the state of current order current_State equal to State.Delivering value (see the if command lines 16 - 19). For the next If condition, the order has delivered, we update current_State = State.Delivered (lines 19-21)

The while command to verify whether the Demander send the remaining or not (see V for more details). Lines 30 and 31 show the Demander paying the remaining amount of money and our process; otherwise, please follow lines 32 and 33. Finally, the Supplier receives their amount (see lines 35-37) as well as the TransactionManager (see lines 38 - 40)

Moreover, the Letter-of-Credit Chain also provides the RBAC service for the three actors in the system. In this service, we allow the authorized partner can call the corresponding function/method. The list of functions is presented in our paper public in [7]. However, the data and meta data of the transaction is still public for all the stakeholders. Please follow our analysis w.r.t security and privacy which describes in Section VII-D

Algorithm 1 Letter-of-Credit Chain Execution

```
1:  Input: contract_Name, Transaction_Manager, Demander,
    Supplier, order_Amount, tax_Amount, deposit_Amount
2:  current_State = State.Manager_Withdrawm
3:  Begin: set balance_Received = 0; set current_State =
    State.Created
4:  while Demander transfers deposit>=deposit_Amount do
    do
5:      update balance_Received
6:      storing payment transaction to payment_Histories
7:      if balance_Received<deposit_Amount then
8:          update current_State = State.DEPOSIT
9:      else if balance_Received>=deposit_Amount && bal-
        ance_Received<order_Amount then
10:         update current_State = State.HOLD
11:     else
12:         update current_State =
            State.COMPLETE_PAYMENT
13:     end if
14:     storing order's status to orderStatus
15: end while
16: if current_State == State.HOLD or current_State ==
    State.COMPLETE_PAYMENT then
17:     manual update current_State = State.SHIPPING
18: end if
19: if current_State == State.SHIPPING then
20:     manual update current_State = State.SHIPPED
21: end if
22: while receiving the rest payment until the deadline do
23:     if balance_Received == order_Amount then
24:         update current_State =
            State.COMPLETE_PAYMENT
25:         manual update current_State =
            State.CAN_WITHDRAW
26:     else
27:         manual update current_State = State.NONPAYMENT
28:     end if
29: end while
30: if current_State == State.CAN_WITHDRAW then
31:     set amount_Withdraw = balance_Received -
        tax_Amount
32: else if current_State == State.NONPAYMENT then
33:     set amount_Withdraw = balance_Received -
        2*tax_Amount
34: end if
35: while Supplier withdrew money do
36:     current_State = State.WITHDRAWN
37: end while
38: while TransactionManager withdrew money do
39:     current_State equal to State.MWITHDRAWN
40: end while
```

VII. Evaluation

A. Environmental Setting

The setting of our environment is shown below:

- Blockchain platform: Binance Smart Chain, ETH, Fantom

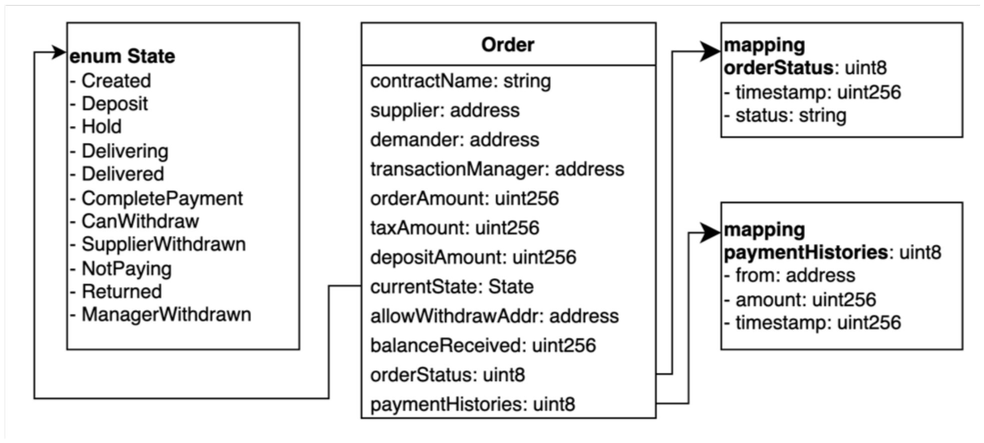

Fig. 3. Data Structure.

- Language: Solidity
- IDE: Remix
- Compiler: 0.8.16+commit.e07a7930a
- Evm version: default
- Gas limit: 3000000
- Optimization: yes
- Open Source License Type: MIT License

B. Results

To prove the effectiveness of Letter-of-Credit Chain, we set up two scenarios (i.e., Supplier pays and not pays the rest of payment) in the three most common EVM Blockchain platforms, namely, Fantom, Ethereum, and Binance Smart Chain. We also provide the source code and installation for the further extension in this topic[4]. We consider the two scenarios (i.e., Demander pay the full payment and Demander does not pay the rest of the payment).

1) Scenario 1: Supplier pays full payment: This scenario will sequentially trigger the states as State.Created, State.Hold, State.Delivering, State.Delivered, State.CompletePayment, State.Can_Withdraw, State.Supplier_Withdrawn, and State.ManagerWithdrawn.[5] In this scenario, the Supplier transferred the payments remaining within the validation time. The result of the first scenario is shown in Table I:

2) Scenario 2: Supplier does not Pay the Rest of Payment: This scenario will trigger the states sequentially as State.Created, State.Hold, State.Delivering, State.Delivered, State.NotPaying, State.Returned, State.Can_Withdraw,State.Supplier_Withdrawn, and State.ManagerWithdrawn. In this scenario, the TransactionManager and the Supplier do not transfer the rest of the payment to the smart contract within the validation time.[6]. The result of the second scenario is shown in Table II.

C. Discussion

Tables I and II present the gas fees for the deployment and execution on the three platforms, i.e., Fantom, Ethereum, and Binance Smart Chain of the Letter-of-Credit Chain[7]. We can easily see that Fantom's smart contract execution fee is the cheapest compared to the other two (i.e., on average 0.08 FTM with $0.02665366). Specifically, the most expensive method is executed with approximately 0.5 FTM ($0.13); whereas the cheapest one is $0.0045 with 0.017 FTM for the two scenarios. On the other hand, ETH is the most expensive, with $8.87 for the most and $0.1 for the least for the two scenarios. Following the ETH is the BSC execution fee of $6.6 to deploy the contract on BSC, which is approximately 17 times higher than Fantom's ones and $0.35 for the cheapest ones for the two scenarios. On average, the gas for all eight functions/methods in the two scenarios is $1.49 and $1.29 for the deployment on Ethereum and Binance Smart Chain, respectively.

D. Security and Privacy Discussion

In this article, we just provide the Letter-of-Credit chain based on the blockchain, which focuses on the decentralize and transparency rather than security and privacy (S&P) issues. In these aspects, we support the basic authorization via the role base access control, in which the right party can be called the corresponding functions. However, the main drawback of the RBAC is that on large-scale systems, RBAC is limited at the #roles. These systems might conflict with or redundancy the new policy; thus, the malicious might attack the system [29]. To this end, we will exploit the attribute-based access control (ABAC) approach, which is introduced by [30] to manage the access control process. In particular, Son et al. [14] define the two-layer of policy for the on-chain and off-chain, respectively. Similarly, some approaches split the original policy into sub-policy (e.g., [31]), i.e., public and private policies to ensure

[4]The implementation/deployment of Letter-of-Credit Chain on:
Fantom platform: https://testnet.ftmscan.com/address/0xF11Fde29e0EB94d977d44c2660F5e0227DC81462#code;
Ethereum platform: https://kovan.etherscan.io/address/0xc3f2e07d850d9131123513e3a106c2ce02b8fa21#writeContract;
Binance Smart Chain platform: https://testnet.bscscan.com/tx/0x236fd512f44fa21148e0f902e72277619e2438d704fe9bfa7d6a8db55f1861b7

[5]see the detail of the function from the previous our publication [7].

[6]see the detail of the function from the previous our publication [7]

[7]redemption value as on 29 August 2022

TABLE I. SCENARIO 1: DEMANDER PAYS FULL PAYMENT

Gas for	Fantom	Ethereum	BNB Chain
Create contract	0.4914263925 FTM ($0.13435057)	0.006000447516801253 ETH ($8.876882043)	0.02384379 BNB ($6.617128601)
Transfer deposits	0.059103118802 FTM ($0.016158143)	0.000422128501969933 ETH ($0.624484242)	0.00266919 BNB ($0.740753609)
Delivering	0.016709903756 FTM ($0.004568304)	0.000079564000556948 ETH ($0.117704596)	0.00078564 BNB ($0.218030813)
Delivered	0.016745106924 FTM ($0.004577928)	0.000079732000558124 ETH ($0.11795313)	0.00078732 BNB ($0.218497046)
Allow Supplier Withdraw	0.027616885296 FTM ($0.007550153)	0.000131498000920486 ETH ($0.194534198)	0.00125798 BNB ($0.34911461)
Supplier Withdraw	0.043235290826 FTM ($0.011820053)	0.000205864001441048 ETH ($0.304549028)	0.00196664 BNB ($0.545781933)
Manager Withdraw	0.027618085404 FTM ($0.007550481)	0.000131503001052024 ETH ($0.194541595)	0.00126903 BNB ($0.352181206)

TABLE II. SCENARIO 2: SUPPLIER DOES NOT PAY THE REST OF PAYMENT

Gas for	Fantom	Ethereum	BNB Chain
Create contract	0.49142639 FTM ($0.13435057)	0.00600045 ETH ($8.87688204)	0.02384379 BNB ($6.61712860)
Transfer deposits	0.05910312 FTM ($0.01615814)	0.00042213 ETH ($0.62448424)	0.00266919 BNB ($0.74075361)
Delivering	0.01670990 FTM ($0.00456830)	0.00007956 ETH ($0.11770460)	0.00078564 BNB ($0.21803081)
Delivered	0.01674511 FTM ($0.00457793)	0.00007973 ETH ($0.11795313)	0.00078732 BNB ($0.21849705)
not Paying	0.02761689 FTM ($0.00755015)	0.00013150 ETH ($0.19453420)	0.00125803 BNB ($0.34912849)
Supplier Withdraw	0.04323529 FTM ($0.01182005)	0.00020586 ETH ($0.30454903)	0.00196664 BNB ($0.54578193)
Manager Withdraw	0.02761809 FTM ($0.00755048)	0.00013150 ETH ($0.19454159)	0.00126903 BNB ($0.35218121)

the data is only accessed via permission even by the parties in the same transaction. Besides, the query rewriting can apply to the complex context where the released data is shared with multiple parties and dynamic context [32], [33]. In particular, the authors proposed the dynamic query VII, which responds the difference value (i.e., details level) based on the user attribute.

VIII. CONCLUSION

This article introduced Letter-of-Credit Chain, which replaces the traditional international trade process based on Letter-of-Credit transactions with edge-cutting ones. Specifically, Letter-of-Credit Chain harnesses the benefits of Blockchain technology and smart contracts to build three user groups: transaction manager, supplier, and demander. Letter-of-Credit Chain's execution process is based on eight steps, from the supplier initiating the order to passing the order on to the demander. Compared to the traditional model of depending on a trusted third party (i.e., a bank), Letter-of-Credit Chain is aimed at decentralized storage, where users can view relevant information regarding their order. If a dispute arises, the role of the transaction manager emerges as the one who decides who should be penalized. Specifically, in the current version, the sanction requirements are all public in the smart contract previously defined by the transaction manager. To demonstrate the effectiveness of the Letter-of-Credit Chain, we implement proof-of-concept on all three popular EVM-enabled platforms today, Fantom, Binance Smart Chain, and Ethereum. The performance comparison section proved that Fantom is a viable platform to deploy for further studies.

Regarding future research directions, we plan to develop the current RBAC model that supports more attributes with the application of ABAC. In addition, the privacy scalability of the Letter-of-Credit Chain is also considered by designing a dynamic request mechanism where the return result depends on the requester's role. On the other hand, Letter-of-Credit Chain does not yet include a shipper role, so a serious consideration to extending the current model is urgently needed.

REFERENCES

[1] S.-Y. Wong and K.-S. Chin, "Organizational innovation management: An organization-wide perspective," *Industrial Management & Data Systems*, 2007.

[2] A. Shleifer and R. W. Vishny, "Liquidation values and debt capacity: A market equilibrium approach," *The journal of finance*, vol. 47, no. 4, pp. 1343–1366, 1992.

[3] J. Dolan, "The law of letters of credit," *THE LAW OF LETTERS OF CREDIT, 4th edition*, pp. 07–36, 2007.

[4] D. Waters, *Supply chain risk management: vulnerability and resilience in logistics*. Kogan Page Publishers, 2011.

[5] H. H. Luong, T. K. N. Huynh, A. T. Dao, and H. T. Nguyen, "An approach for project management system based on blockchain," in *International Conference on Future Data and Security Engineering*. Springer, 2021, pp. 310–326.

[6] X. S. Ha, H. T. Le, N. Metoui, and N. Duong-Trung, "Dem-cod: Novel access-control-based cash on delivery mechanism for decentralized marketplace," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2020, pp. 71–78.

[7] K. L. Quoc, H. K. Vo, L. H. Huong, K. H. Gia, K. T. Dang, H. L. Van, N. H. Huu, T. N. Huyen, L. Van Cao Phu, D. N. T. Quoc *et al.*, "Sssb: An approach to insurance for cross-border exchange by using smart contracts," in *International Conference on Mobile Web and Intelligent Information Systems*. Springer, 2022, pp. 179–192.

[8] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[9] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "A survey on the adoption of blockchain in iot: Challenges and solutions," *Blockchain: Research and Applications*, vol. 2, no. 2, p. 100006, 2021.

[10] M. Alharby and A. Van Moorsel, "Blockchain-based smart contracts: A systematic mapping study," *arXiv preprint arXiv:1710.06372*, 2017.

[11] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

[12] S. Shi, D. He, L. Li, N. Kumar, M. K. Khan, and K.-K. R. Choo, "Applications of blockchain in ensuring the security and privacy of electronic health record systems: A survey," *Computers & security*, vol. 97, p. 101966, 2020.

[13] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran, "An overview on smart contracts: Challenges, advances and platforms," *Future Generation Computer Systems*, vol. 105, pp. 475–491, 2020.

[14] H. X. Son *et al.*, "Towards a mechanism for protecting seller's interest of cash on delivery by using smart contract in hyperledger," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 4, pp. 45–50, 2019.

[15] X. S. Ha, T. H. Le, T. T. Phan, H. H. D. Nguyen, H. K. Vo, and N. Duong-Trung, "Scrutinizing trust and transparency in cash on delivery systems," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer, 2020, pp. 214–227.

[16] N. Duong-Trung, H. X. Son, H. T. Le, and T. T. Phan, "Smart care: Integrating blockchain technology into the design of patient-centered healthcare systems," in *Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy*, ser. ICCSP 2020, 2020, p. 105–109.

[17] ——, "On components of a patient-centered healthcare system using smart contract," in *Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy*, 2020, p. 31–35.

[18] H. T. Le, K. L. Quoc, T. A. Nguyen, K. T. Dang, H. K. Vo, H. H. Luong, H. Le Van, K. H. Gia, L. V. Cao Phu, D. Nguyen Truong Quoc *et al.*, "Medical-waste chain: A medical waste collection, classification and treatment management by blockchain technology," *Computers*, vol. 11, no. 7, p. 113, 2022.

[19] H. X. Son, T. H. Le, N. T. T. Quynh, H. N. D. Huy, N. Duong-Trung, and H. H. Luong, "Toward a blockchain-based technology in dealing with emergencies in patient-centered healthcare systems," in *International Conference on Mobile, Secure, and Programmable Networking*. Springer, 2020, pp. 44–56.

[20] H. T. Le, L. N. T. Thanh, H. K. Vo, H. H. Luong, K. N. H. Tuan, T. D. Anh, K. H. N. Vuong, H. X. Son *et al.*, "Patient-chain: Patient-centered healthcare system a blockchain-based technology in dealing with emergencies," in *International Conference on Parallel and Distributed Computing: Applications and Technologies*. Springer, 2022, pp. 576–583.

[21] N. T. T. Quynh, H. X. Son, T. H. Le, H. N. D. Huy, K. H. Vo, H. H. Luong, K. N. H. Tuan, T. D. Anh, N. Duong-Trung *et al.*, "Toward a design of blood donation management by blockchain technologies," in *International Conference on Computational Science and Its Applications*. Springer, 2021, pp. 78–90.

[22] H. T. Le, T. T. L. Nguyen, T. A. Nguyen, X. S. Ha, and N. Duong-Trung, "Bloodchain: A blood donation network managed by blockchain technologies," *Network*, vol. 2, no. 1, pp. 21–35, 2022.

[23] H. T. Le *et al.*, "Introducing multi shippers mechanism for decentralized cash on delivery system," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, 2019.

[24] Ethereum, "How our escrow smart contract works," 2022. [Online]. Available: https://www.thenational.ae/business/technology/cash-on-delivery-the-biggest-obstacle-to-e-commerce-in-uae-and-region-1

[25] OpenBazaar, "Truly decentralized, peer-to-peer ecommerce features," 2022. [Online]. Available: https://openbazaar.org/features/

[26] "Two party contracts," 2022. [Online]. Available: https://dappsforbeginners.wordpress.com/tutorials/two-party-contracts/

[27] N. Duong-Trung *et al.*, "Multi-sessions mechanism for decentralized cash on delivery system," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 9, 2020.

[28] N. T. T. Le *et al.*, "Assuring non-fraudulent transactions in cash on delivery by introducing double smart contracts," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 5, pp. 677–684, 2019.

[29] N. M. Hoang and H. X. Son, "A dynamic solution for fine-grained policy conflict resolution," in *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, 2019, pp. 116–120.

[30] H. X. Son and N. M. Hoang, "A novel attribute-based access control system for fine-grained privacy protection," in *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, 2019, pp. 76–80.

[31] Q. N. T. Thi, T. K. Dang, H. L. Van, and H. X. Son, "Using json to specify privacy preserving-enabled attribute-based access control policies," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer, 2017, pp. 561–570.

[32] H. X. Son, T. K. Dang, and F. Massacci, "Rew-smt: a new approach for rewriting xacml request with dynamic big data security policies," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer, 2017, pp. 501–515.

[33] S. H. Xuan *et al.*, "Rew-xac: an approach to rewriting request for elastic abac enforcement with dynamic policies," in *2016 International Conference on Advanced Computing and Applications (ACOMP)*. IEEE, 2016, pp. 25–31.