The Trust Chain Consensus

COTI: a decentralized, high performance cryptocurrency ecosystem optimized for creating digital payment networks and stable coins

Technical White Paper, V.4.0, 2nd October, 2018

Abstract

It has been long anticipated for cryptocurrencies to significantly alter the online payment landscape. To accomplish this, it is necessary for cryptocurrencies to be user friendly, convenient and highly scalable. Several blockchain based technologies have been created to tackle the challenges posed by attempting to provide high transaction throughput while remaining inexpensive, but these have been met with little success. Another challenge faced is the lack of trust between unknown parties, which leads to countless chargebacks and transaction cancellations. Moreover, merchants are often classified as 'high-risk' or 'low-risk' based on their association with a particular industry, rather than on their actual behaviour.

COTI, Currency Of The Internet, solves these challenges by using an innovative base-layer protocol in the form of a directed acyclic graph-based ledger, built from transactions connected through time, by harnessing trust-based algorithms. Trust should be based on a combination of the network participant's historical behaviour data and objective information about them. COTI takes this into consideration, calculating trust using a unique machine-learning algorithm. Trust is used in the Trustchain Algorithm to validate and confirm transactions faster. Trust chains grow as new transactions attach to two prior transactions which have similar degrees of trust as themselves. This results in an innovative consensus-based confirmation mechanism, where every user is incentivised to have a high level of trust while engaging in trust-building behaviour due to the benefits associated with having a high level of trust (i.e. faster confirmation times).

COTI has built mechanisms to monitor, detect and defend against possible attacks, ensuring network security. An example of such a mechanism is COTI's Double Spend Prevention (DSP) Nodes. COTI also introduces novel protocols to address disputes that may arise when sending transactions, a much required feature which is not possible with other cryptocurrencies. Dispute resolution is achieved by the use of an Arbitration Service. This service takes advantage of the principles of game theory to ensure a fair outcome in the case of a dispute and votes to determine which of the two disputing parties is right.

Our vision of COTI is to empower users to freely exchange value as simply as information is exchanged on the Internet. To achieve this, we are developing the Trustchain Protocol based on a directed acyclic graph (DAG) distributed ledger, which creates a scalable blockless protocol that can be utilised by any industry that needs high throughput and trust to operate.

COTI is uniquely positioned to provide the infrastructure needed for industries requiring immense scalability, in addition to an arbitration mechanism to resolve disputes, fraud (e.g. double spending) and errors.

COTI also introduces a MultiDAG structure and high-performance smart contracts, which provide a multitude of tools for enterprises, merchants, governments, developers and stable coin issuers.

Keywords: Arbitration, Blockchain, COTI, Cryptocurrency, DAG, Distributed ledger, E-commerce

1 Introduction

Blockchain technologies and cryptocurrencies have become alternative mechanisms for managing payment transactions over the past years. Digital currencies such as Bitcoin, Ethereum, and many

others have enjoyed exponential growth in popular interest and adoption[1], while other uses have included technological applications, ranging from supply chain management [11] to decentralised, verifiable health records [17]. Indeed, many have likened cryptocurrencies to the early internet, citing its enormous potential to disrupt payment systems in the same way the internet disrupted information access [14].

However, while first generation cryptocurrencies have been enormously successful, they have faced fundamental challenges that have prevented them from achieving universal adoption. Linear blockchain based cryptocurrencies suffer from low transaction throughput[2]; cryptocurrencies that rely on a network of miners to perform increasingly complex proof-of-work (PoW) computations incur prohibitively high transaction fees; and most existing cryptocurrencies are difficult to manage and are subject to mass speculation. In addition, services like dispute resolution, which are commonplace for credit cards and other payment platforms, are rare within the frameworks of most existing cryptocurrencies. These factors make it difficult for individuals and merchants to adopt them as a global currency or digital dollar for day-to-day transactions.

This paper introduces COTI (Currency Of The Internet), a next-generation cryptocurrency that achieves high transaction throughput and low fees, while being easy to manage and providing decentralised structures for the services users have come to expect from payment platforms, such as dispute resolution. COTI achieves a high transaction throughput by employing a Directed Acyclic Graph (DAG) of transactions known as the Cluster, as opposed to a blockchain. This idea is not new, and has been proven to improve performance [13, 4, 15]. Typically, DAG based cryptocurrencies[3] have been intended for large numbers of low valued transactions, possibly between machines such as IoT devices. Because COTI is designed to support day-to-day transactions between merchants and consumers, new algorithms have been introduced to drive the formation of the Cluster, and the approval of transactions. Fundamental to the new approach is the Trust Score, which is assigned to each user account based on its historical behavior, and which governs the approval of that account's transactions within the network as well as the amount of any possible fees incurred. These algorithms will be described in detail in Section 2.

In addition to the new features mentioned above, COTI introduces an Arbitration Service for dispute resolution, consisting of a decentralised collective of highly trusted network participants who vote on dispute rulings. This allows the network to offer decentralised human-input services to its participants.

The Base Layer Protocol: DAG-based distributed ledger technologies show signs of being particularly adept at overcoming the scalability limitations inherent in blockchain-based payment networks. This is because while in blockchain-based networks, greater scale has undesirable effects on network usability, in DAG-based networks the reverse is generally true: greater network usage results in improved network scalability. In other words, there is a positive correlation between the number of network users and the rate at which transactions are confirmed.

As a result of the positive correlation between network usage and network scalability, the DAG data structure is ideally suited for the COTI network's base layer protocol, and will enable it to achieve full decentralisation without compromising COTI's commitment to scalability, instantaneity and low (or zero) fees. Building on the foundations established by the above-mentioned initiatives, COTI is introducing an innovative DAG-based distributed ledger technology as its base layer protocol, which involves the use of Trust Scores as the key mechanism by which new, unconfirmed transactions select prior transactions to validate. Furthermore, COTI's DAG-based distributed ledger technology, the Cluster, reaches faster consensus when confirming transactions by using COTI's Trustchain Algorithm. Eventually, the Cluster will be able to validate and confirm a maximum of hundreds of thousands transactions per second (TPS)[4].

[1] The number of Bitcoin transactions per day has grown from about 100 in 2009 to over 400,000 in late 2017 [2].
[2] Bitcoin delivers a maximum of 7 transactions per second [6].
[3] e.g. IOTA
[4] Arguments for this can be found in Section 10.

Proof of Trust (PoT): COTI's unique combination of the Trustchain Algorithm and Proof of Work.

COTI coin: COTI introduces a high-performance cryptocurrency built atop the base layer protocol. This cryptocurrency will be used as a common means of payment, including all fees and staking inside the COTI ecosystem.

MultiDAG: COTI is not bounded to one instance of DAG. The same infrastructure of nodes permits the creation of multiple DAGs that can be used for various purposes and originators. There can be voucher tokens, stable coins, dedicated tokens for global companies, or scalability tokens to speed up settlements in other blockchains. See more details in "MultiDAG" section below.

Smart contracts: COTI introduces "on-chain" smart contracts for the DAG, a first of its kind. See more details in "MultiDAG" section below.

The Arbitration Service: COTI offers a ready-to-use service that users can appeal to in cases of fraud or any other dispute related to deals settled through the COTI payment system. The Arbitration Service creates a rolling reserve for each merchant to cover possible claims and a system-wide Reserve Credit Fund (RCF) to guarantee it. Both funds are maintained in COTI's native currency. The required size of a merchant's rolling reserve is calculated based on the merchant's Trust Score. Please refer for details to Appendix B, "COTI's Arbitration System".

Fees: The COTI network uses a transparent and equitable fee model. All fees are collected by Full Nodes (decentralised servers run by ordinary users in the COTI network). The COTI network receives a portion of fees collected by Full Nodes to support infrastructural technology such as the Double Spend Prevention Nodes (see Sections 6.2 and 11.1) and Trust Score Servers (see Section 3.3). When the network is first created, a portion of all generated COTIs will be set aside as a Reserve Fund to pay for all transactions until the network matures. Therefore, the network fee will be set to zero during the network's infancy. Following this period, the fees will be minor due to the decentralised nature of the network.

Each node charges a fee that is in part determined by the node itself. Some nodes may set a higher fee if they believe they provide a good service; other nodes may charge less or possibly nothing. The price charged by a node for its services should be equitable, publicly available and compliant with common network rules. Network rules will define a ceiling for fees, but there will be no minimum fee.

All fees within the COTI ecosystem are paid with the COTI coin.

It is possible for a merchant to run their own Full Node along with a customised wallet if they believe it will provide a better experience for their customers.

Glossary

Term	Meaning
Node	A specialised server run by a user for common network tasks
Transaction validation	The process of checking the transaction before attachment to the Cluster
Source transaction	A terminal transaction of the Cluster having no inbound transactions. The mandatory validation of two prior transactions has been fulfilled for these transactions
Confirmed transaction	A transaction for which the consensus algorithm has reached a defined level of total trust.
Trust Score	A user metric that is used for effective transaction processing and risk mitigation.
Proof of Trust (PoT)	COTI's unique combination of the Trustchain Algorithm and Proof of Work
Attack	A malicious attempt to compromise a system's integrity.
Double-spending	An attack in which the attacker tries to process two transactions using the same account balance. This results in a negative balance and the attacker acquiring something without cost

2 The Trust Chain Algorithm

COTI has developed a new approach to achieving consensus between transacting parties operating on a DAG based data structure. The Cluster is based on a completely decentralised DAG, which is not stored by any central authority. It is a ledger, or a record of all transactions performed by the network. The Cluster achieves scalability through its use of parallel source selection and confirmation of transactions, as well as its use of COTI's Trust Scores.

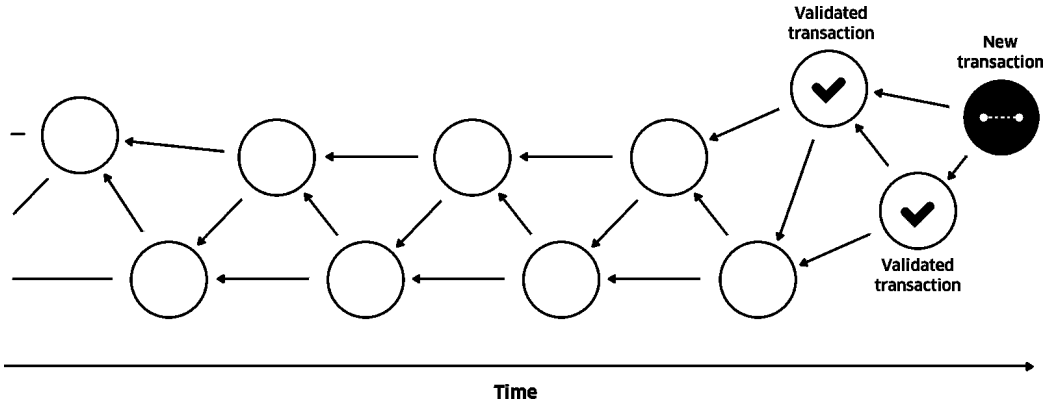
2.1 Trust Score of transactions in the Cluster

Each transaction in the Cluster receives a Trust Score equal to the sending account's Trust Score (further information on the Trust Score Algorithm can be found in the Trust Score Section). A transaction's Initial Trust Score is used to define:

- The unvalidated transactions (Sources) which will be validated and referenced by the transaction (see Source Selection subsection).
- The amount of proof-of-wok (PoW) that should be performed prior to transaction attachment.

As a result, transactions from highly trusted senders are confirmed much faster (please refer to the Performance Investigation section for further details). When attaching a transaction, the Full Node is required to validate two prior transactions in order for the new transaction to be added to the ledger. The ledger is therefore organised as a DAG (directed acyclic graph), where the vertices are transactions and directed edges are drawn from each transaction to two others that it validates. A schematic of the Cluster is shown in Figure 1. Each white circle represents a transaction that has been validated by two subsequent transactions, while the darker circle represents a new, unvalidated transaction i.e. a 'source' in graph-theoretic terminology. As new transactions are added, they may validate the darker transaction.

Figure 1: Cluster schematic. The source transaction (dark circle) validates two previous transactions in the Cluster.



2.2 Source Selection

The process outlined above requires each new transaction to pick two prior source transactions to validate. In COTI, the algorithm for making this selection is based on each transaction's Trust Score. According to this Source Selection Algorithm, a source will likely choose prior transactions which are close to its current Trust Score. This results in the formation of Trustchains in the Cluster. A Trustchain is any reverse-oriented path in the Cluster. The cumulative Trust Score of such a chain is the sum of the Trust Scores of all the transactions making up the chain.

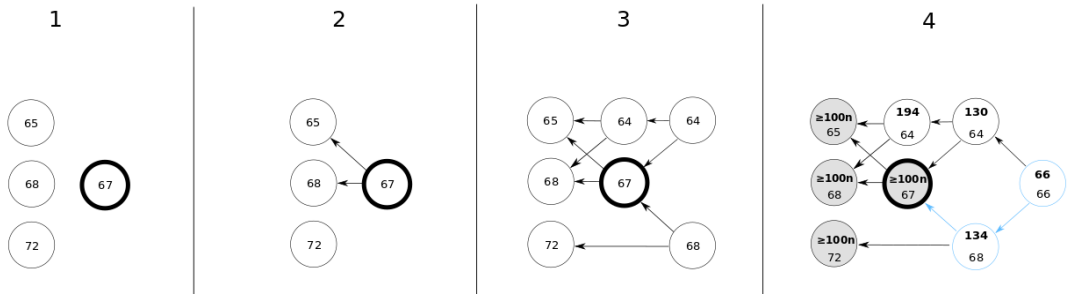
The Trustchain Algorithm makes use of the cumulative Trust Score to achieve consensus on whether a transaction is confirmed or not. A transaction is considered confirmed if it is the starting point of a Trustchain that has a cumulative Trust Score exceeding the pre-set global confirmation threshold. In practice, we consider the longest (highest trust) Trustchain starting from each transaction and compare its cumulative Trust Score to the threshold in order to determine if the transaction has been confirmed.

Because the Source Selection Algorithm tends to connect transactions of similar Trust Scores together, Trust chains generated by highly trusted users will mostly contain transactions with high Trust Scores. The cumulative Trust Score of such a Trustchain will grow quickly past the threshold and achieve consensus, meaning that highly trusted users will enjoy fast confirmation times and high transaction throughput.

Another important consequence of the Source Selection Algorithm is the soft segmentation of the Cluster based on Trust Scores. In other words, DAG sections with different Trust Scores are processed in parallel, while distant DAG sections are essentially independent.

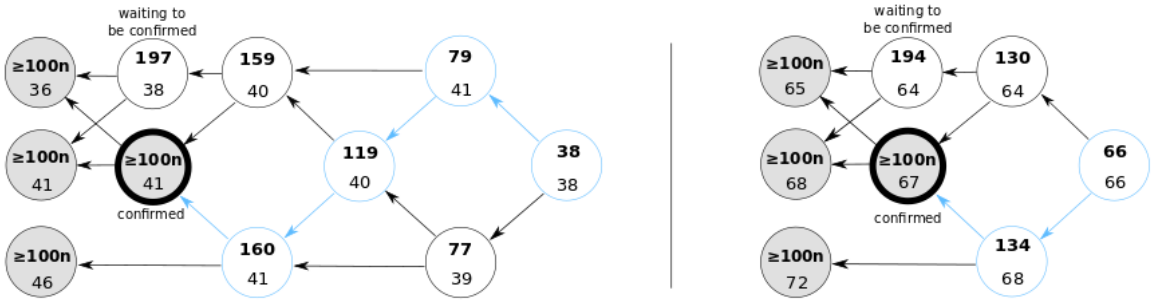
Foregoing any complications, every transaction in the Cluster will progress through the following life stages: in the first stage, it is initiated as a new transaction; in the second, it attaches to the Cluster by validating two other transactions with Trust Scores similar to its own; in the third, it is validated by other transactions; finally, it is confirmed and permanently added to the Cluster once the cumulative Trust Score of the heaviest path confirming it surpasses the set threshold. This process is illustrated in Figure 2

Figure 2: The lifecycle of a transaction (bold circle) from (1) initiation, to (2) attachment, to (3) validation, to (4) confirmation. The confirmation path is in blue, while the shaded transactions have been confirmed. For the purposes of this illustration $n = 2$.



COTI's Trustchain Algorithm is designed in such a way that trusted users (i.e. those with a high Trust Score) will experience faster confirmation times than those who are less trustworthy. This is expectable as people are more cautious when dealing with people they trust less and so would like to be more certain that their transactions are confirmed before accepting them. This property is illustrated in Figure 3 and further discussed using simulations in section 10.

Figure 3: The different Trust Chain lengths needed to confirm moderately trusted transactions (left) and highly trusted transactions (right). Shaded transactions are those that have reached the cumulative trust threshold, while the confirmation path is in blue. Shaded transactions have been confirmed. For the purposes of this illustration $n = 2$.



2.3 Source Selection Algorithm

The primary objective is to build a Cluster based on the transaction sender's Trust Score. In the Cluster, every transaction is attached to two transactions at most with Trust Scores that are sufficiently close to its own. On the DAG, the Trust Score of the transaction sender is assigned to each transaction with a weight function ω. Let d be the upper bound for Trust Scores. According to the Trust Score Algorithm, this is equal to 100.

Any good method for constructing such a DAG must be based on an algorithm that chooses two transactions with some degree of randomness. Consider, for example, an algorithm which chooses two transactions b and c based only on having Trust Scores closest to the Trust Score of the source a. One can see that the use of a non-random algorithm such as the one just described, increases the probability of many sources being present that must wait a long time to be attached by a transaction.

COTI's Source Selection Algorithm works in the following way: a new transaction a is issued by an account, while S is the set of all the sources. The algorithm chooses the optimal neighbourhood of $\omega(a)$. First, all the sources are partitioned with a map function $M : \{1; 2; ; d\} \to \{T : T \subseteq S\}$ such that $M(i) = \{T : T \subseteq S \text{ and } \omega(T) = i\}$, where d is the upper bound for the Trust Score. The initial subset is $T_0 = M(\omega(a))$. Iterations in the algorithm generate new subsets $T_i = T_{i-1} \cap M(\omega(a) - i) \cap M(\omega(a) + i)$ until T_i is sufficiently populated, or $i < [d/8]$. Without any loss of generality, it can be said that being sufficiently populated connotes a constant percentage of all the sources in the source set. 10% of the source population is chosen to be sufficiently populated. If at any iteration $\omega(a) - i < 0$ or $\omega(a) + i > d$,

then $M(\omega(a) - i)$, or $M(\omega(a) + i)$, respectively, is taken to be the empty set. There is one further restriction that must be applied to subset T_j of sources in the neighbourhood of $\omega(a)$, namely that no transaction may be attached to a transaction with the same transaction sender. A probability function P weighting all sources s in T_j according the timestamp difference between s and new transaction a is defined by the algorithm. a can then select any two sources in T_j with some degree of randomness, but such that the older sources will be chosen with a higher probability than the newer sources. There is zero probability of selecting sources from the same transaction sender of a. The Java code in Algorithm 1 shows how this is done.

Algorithm 1: Java code showing how sources are selected.

```
public SourceList selectSources(int trustScore,
int minSourcePercentage, int totalSourceNum, int maxNeighbourhoodRadius) {

    // Start by taking the sources with the same Trust Score (clone)
    SourceList sourceList = new SourceList(sourcesByTrustScore.get(trustScore));

    // Calculate the neighbourhood radius, minimal radius is 1 (always look at neighbours)
    for(int nr=1; nr < maxNeighbourhoodRadius; nr++) {

        if(trustScore - nr >= 1)
            sourceList.add(sourcesByTrustScore.get(trustScore - nr));
        if(trustScore + nr <= MAX_SCORE )
            sourceList.add(sourcesByTrustScore.get(trustScore + nr));

        if((double)sourceList.size() / totalSourceNum > (double)minSourcePercentage / 100) {
            break;
        }
    }

    // Randomly choose source, weighted by timestamp difference
    return chooseWeightedByTimestamp(sourceList);
}
```

Note that our algorithm can respond to changes in the flow of new transactions since it takes into account the number of sources in a transaction's neighbourhood. In the Cluster's early stages, there will be cases when transaction a cannot be attached to any transaction in T_j (e.g. when all the source Trust Scores are accumulated too far from $\omega(a)$, or when all the sources in the selected neighbourhood are from the same transaction sender as a).

In these cases, a Zero Spend Server will create a zero-value transaction with the same Trust Score of transaction a and a will be attached to that transaction. In another scenario, if a source s is waiting a long time to be attached to by a new transaction, then a Zero Spend Server will create a transaction to attach to s with the same Trust Score as s. The waiting time before the Zero Spend Server performs these tasks will be determined according to the Trust Score: high Trust Score sources will be matched faster by the Zero Spend Servers. As mentioned in Section 6.3, over-activity of Zero Spend Servers will help to identify problems in the network or in the Trust Score Algorithm.

2.4 Attachment Process

The following steps will be performed when a new transaction is received from a wallet:

- Address validation
- Balance check
- Pre-balance check
- Source selection
- Source validation
- Proof of work (PoW)
- DAG attachment

- Propagation to other Nodes

First, the transaction is validated. The addresses of each base transaction and of the entire transaction are checked. Then the balance is checked to verify that each address has sufficient funds.

The pre-balance is then checked to prevent double spending from the same wallet. Then the process of source selection, validation, PoW and attachment to the local DAG are performed. After the transaction is locally validated, tested and attached in a Full Node, it is propagated to other Nodes and receives Trustchain Consensus and DSP Consensus (see Double Spend Prevention and DSP Consensus for further details).

Upon receiving the transaction, other Nodes do not perform any PoW processing or other consistency verifications for the transaction. The transaction is checked when other Nodes attach new transactions to it according to the source selection algorithm. The transaction will be declined if the transaction fails these checks.

Transactions cannot be approved by other transactions which were initiated by the same user(seed) or created by the same Full Node.

2.5 Trustchain consensus

Let $G = (V; E)$ be a DAG of transactions. Assume that every transaction $v \in V$ is weighted with a weight function $\omega : V \to N$ defined by $\omega(v)$, the Trust Score of transaction sender. Let a be any transaction and d be the upper bound for the Trust Score. We can say the transaction a is confirmed if

$$max(\Sigma_v \omega(v) : \forall \text{ path A ended at } a \text{ and } \forall v \in A)Ld, L \geq 2 \tag{2.1}$$

Equation 2.1 implies that highly trusted transactions will be confirmed faster than less trusted transactions due to the Source Selection Algorithm. Note that a highly trusted transaction a is confirmed quickly largely because the length of the heaviest directed path is very small. Notice also that less trusted transactions need a longer path to be confirmed. The algorithm for the heaviest directed path which ends at transaction a is a linear time algorithm, namely $O(|V(F)| + |E(F)|)$ where $F \subset G$ is a directed acyclic subgraph defined by the union of all directed paths ending at transaction a. The first step is to sort F topologically. Since F is a DAG, finding a topological sort τ is linear time (Chapter 22.4 in [5]). Let $\tau = v1; v2; \ldots; vn; a$ be a topological sort of F. Notice that transaction a should be the last vertex at the topological sort due to the definition of F.

Figure 4: DAG subgraph F before topological sorting

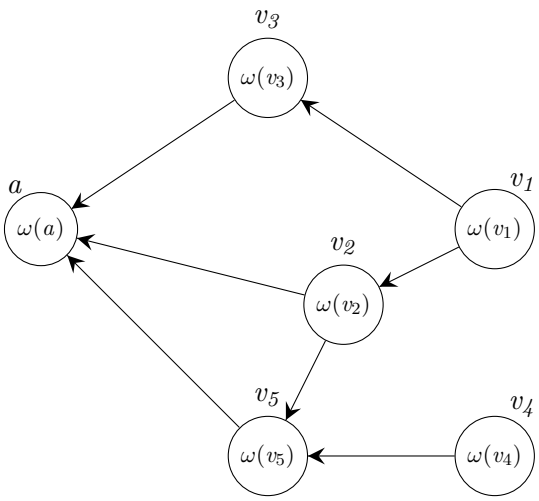
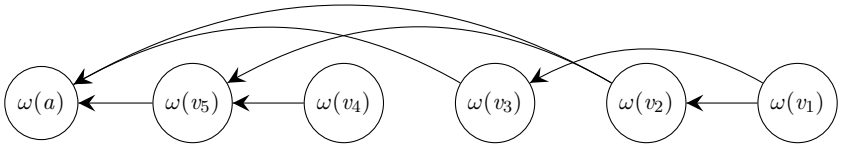


Figure 5: DAG subgraph F after topological sorting



The following dynamic programming algorithm gives $O(|V(F)| + |E(F)|)$ time solution for the heaviest path from transaction a:

Algorithm 2: Heaviest Path Algorithm

```
Define function heaviest(v) = 0 , ∀v ∈ V(F);
Find Topological sort τ of F;
for ∀v ∈ V(F) in topological sort τ do
    Assign heaviest(v) = max (heaviest(w) + ω(v): ∀w such that (w, v) ∈ E(F));
end
return heaviest(a);
```

2.6 Balance Check

In order to keep the consistency of the payment system and double spend prevention, COTI's Full Nodes and DSP Nodes carry out balance checks. When Full Nodes and DSP Nodes are running, all address balances are calculated. When a Node is restarted, balances are recalculated from the last Clusterstamp state. There are two types of balances controlled by Full Nodes and DSP Nodes: Current balance: consists of both Trust Chain consensus and DSP-confirmed transactions with positive address balances Pre-balance: consists of all verified transactions. COTI Full Nodes and DSP Nodes check balances independently, providing DSP consensus. For details, please refer to the Double Spend Prevention and DSP Consensus section.

For different Clusters in the MultiDAG, various balance check approaches can be implemented.

3 MultiDAG

3.1 An additional dimension of the protocol

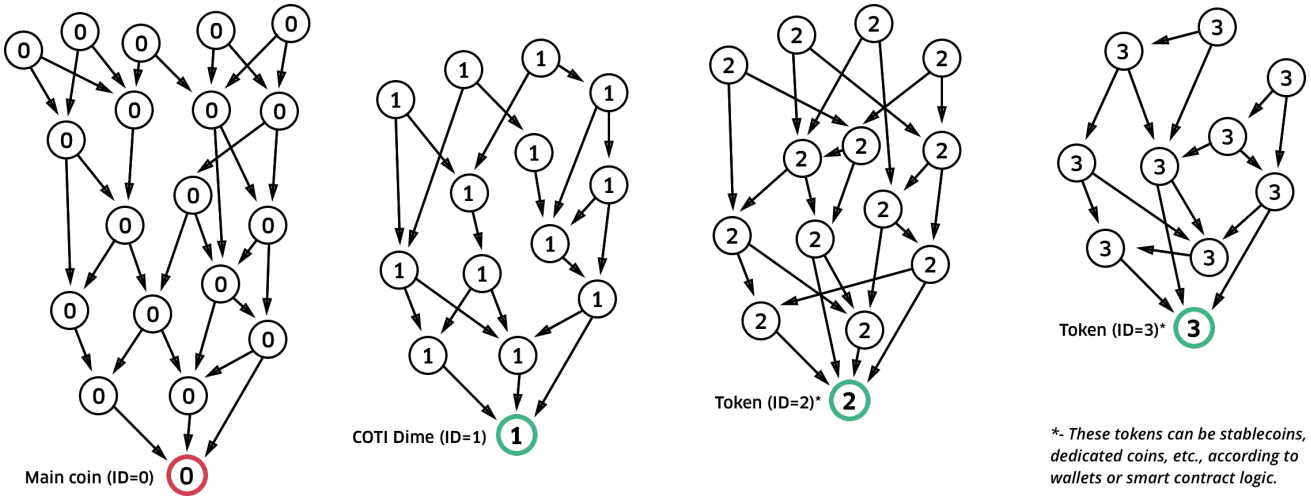
The DAG model provides COTI with the framework for exceptional performance and efficient transaction processing. There are multitudes of token types and uses, which are worthwhile to implement atop the DAG.

COTI uses several independent Clusters that each support one token, which makes the whole network efficient and adjustable. A transaction in a Cluster can be attached to transactions in the same Cluster because different Clusters can implement various transaction confirmation rules.

The COTI MultiDAG ecosystem is similar to that of Ethereum. A common decentralized infrastructure is the basis for many different tokens and smart contracts and one main coin for fee payments.

As per the terms defined above, COTI has created several Clusters for different uses.

Figure 6: Multiple Clusters (DAGs) upon the same infrastructure



On the picture above, the Trustchain structure of the Clusters is omitted for simplicity.

In the COTI MultiDAG, different Clusters are separable at the transaction layer by the ClusterID. For simplicity, in this paper we use integer numbers as ClusterIDs.

The Cluster with ClusterID=0 represents the COTI coin. The difference between ClusterID=0 and others is important: the genesis transaction for ClusterID=0 is created when the network is launched and cannot be added after. Genesis transactions for other Clusters are created by the Cluster owner or smart contracts according to the particular Cluster rules.

Transactions in the ClusterID=0 are confirmed, taking into account the balances of all relevant users. Transaction confirmations in other Clusters can implement other rules.

All fee collection in COTI is the responsibility of Full Nodes. For this reason, Full Nodes must create fee paying transactions in the main coin DAG before transaction attachment.

All Clusters in the COTI MultiDAG use the same identification and KYC procedures, which creates one common frictionless crypto universe that includes many tokens of various types.

3.2 Smart contracts

This section briefly introduces COTI's **smart contracts**.

Please refer to the COTI public github repository for further details.

COTI's smart contract were inspired by Ethereum, an industry standard de-facto and invaluable theoretical basis[5]. Similar to Ethereum smart contracts, COTI smart contracts provide Turing-complete computational models.

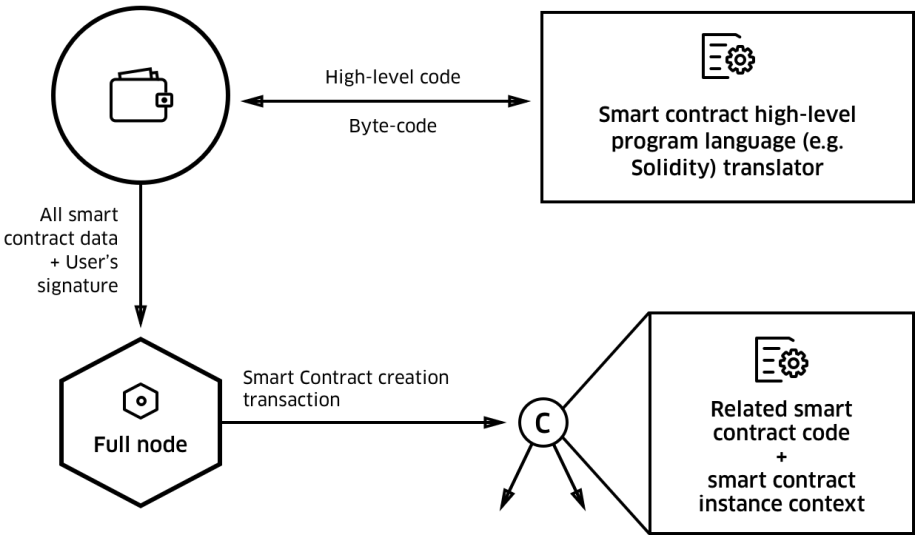
Unlike most other high-performance projects, where smart contracts are executed off-chain using specialized servers or nodes, in a quasi-decentralized manner, COTI smart contracts are executed on-chain and decentralized. All steps of smart contract execution are recorded in the dedicated Cluster in the COTI MultiDAG and are verified several times by various Full Nodes before receiving full confirmation (see "The Trust Chain Algorithm" section above for details of attachment and confirmation process).

COTI smart contracts are created and signed by a COTI user from the wallet application. Smart contracts are coded using specialized high-level program language (e.g. Solidity) and translated to

[5]https://github.com/ethereum/wiki/wiki/Design-Rationale

low-level VM-executable language (byte code). Smart contracts are stored as a transaction in the smart contract Cluster with an address specifying the execution context.

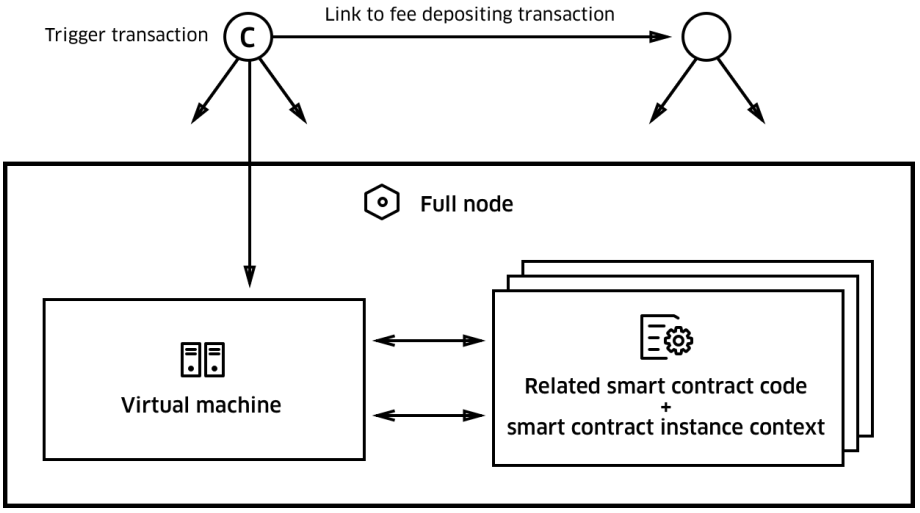
Figure 7: COTI smart contract creation



For simplicity, all fees are omitted above.

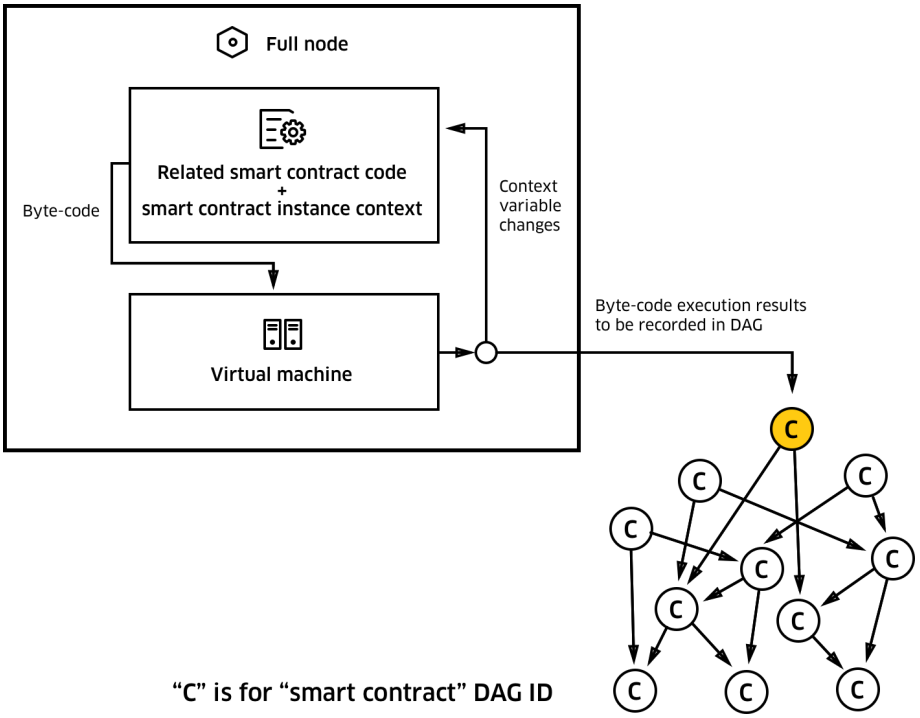
To be re-enterable, created smart contracts are started to execution by a trigger transaction, which defines the particular instance of a launched smart contract. Trigger transactions also provide a link (hash) to the transaction depositing COTI coins in order for the smart contract fee to be paid.

Figure 8: The start of COTI smart contract execution



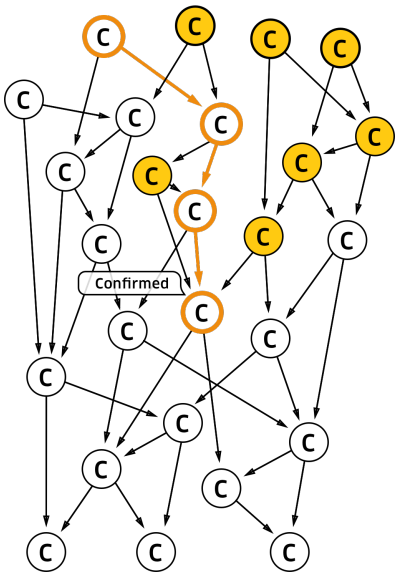
The COTI smart contract virtual machine is part of the standard COTI Full Node code (to be implemented in the advanced TestNet). Upon executing the smart contract bytecode, the VM changes smart contract context variables internal to each Full Node and records the result as a new transaction in the smart contract Cluster.

Figure 9: COTI smart contract execution cycle



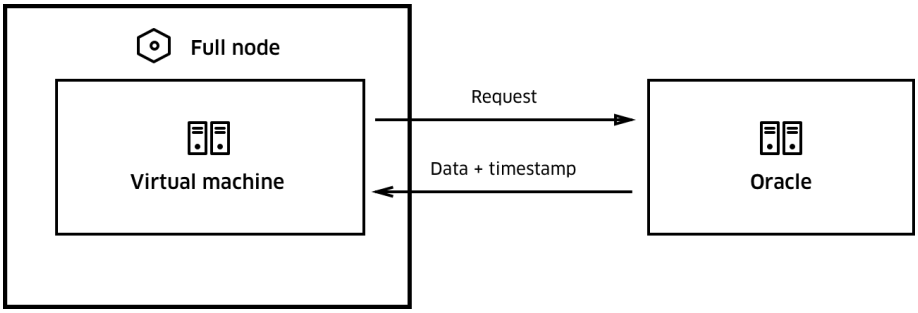
Smart contract bytecode is executed in parallel by all Full Nodes. For a new transaction to be attached to the smart contract Cluster, Full Nodes have to verify two previous transactions. The same applies for all Cluster transactions, as a smart contract execution transaction is considered to be confirmed after the heaviest path from the transaction to the Cluster's fringe reaches the confirmation threshold (see "The Trust Chain Algorithm" section above for details). If the transaction for the bytecode instruction is already attached, the Full Node checks the results and adds its signature. Full Node smart contract transaction verifications affect the Full Node's Trust Score.

Figure 10: Smart contract Trust Chains



To make decentralized consensus on smart contract execution possible, it should be completely deterministic. This entails that smart contracts only be used for on-chain data. Any real world data should be supplied to the calculation process only using oracles while providing data with the corresponding timestamp.

Figure 11: Requesting data from an oracle



Due to COTI's blockless structure (DAG), the network doesn't require gas conception. Fees for COTI smart contract execution are fixed to be minimal and economical. The execution fee for bytecode instructions belongs to the Full Node attached to the transaction and includes the execution fee. As for transaction network fees, they must be transferred to the Network Pool.

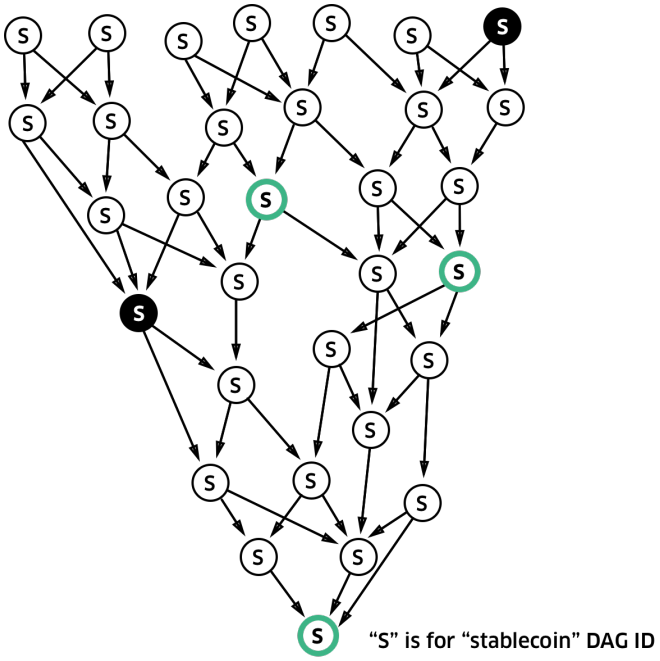
3.3 Stablecoins

COTI's MultiDAG, together with COTI smart contracts and the possibility of multiple genesis transactions allows for the creation of high performance stable coins.

Most stable coins are essentially ERC-20 tokens, which make them hardly usable for everyday payments. For COTI, the situation is different as high throughput, quick confirmations and low fees enable the creation of highly usable stable coins.

In COTI, each stable coin's transactions constitute its own Cluster with its own confirmation rules. All stable coin Clusters are organized according to the transaction sender's Trust Score.

Figure 12: Stable coin Cluster



A stable coin is a cryptocurrency with a constant rate to some real-world asset. It means that the supply of a stable coin cannot be fixed. Stable coin tokens are minted and burned according to market movements.

In the picture above, genesis or token minting transactions are green and token burning transactions

are black. The minting and burning of transactions is created by smart contracts or the stable coin originator's wallet according to the particular stable coin's rules.

The COTI MultiDAG allows for the origination of stable coins of all known types: fiat collateralized, gold (or other asset) collateralized, crypto collateralized and non-collateralized. Stable coins can be originated by COTI itself or, more commonly, by third parties.

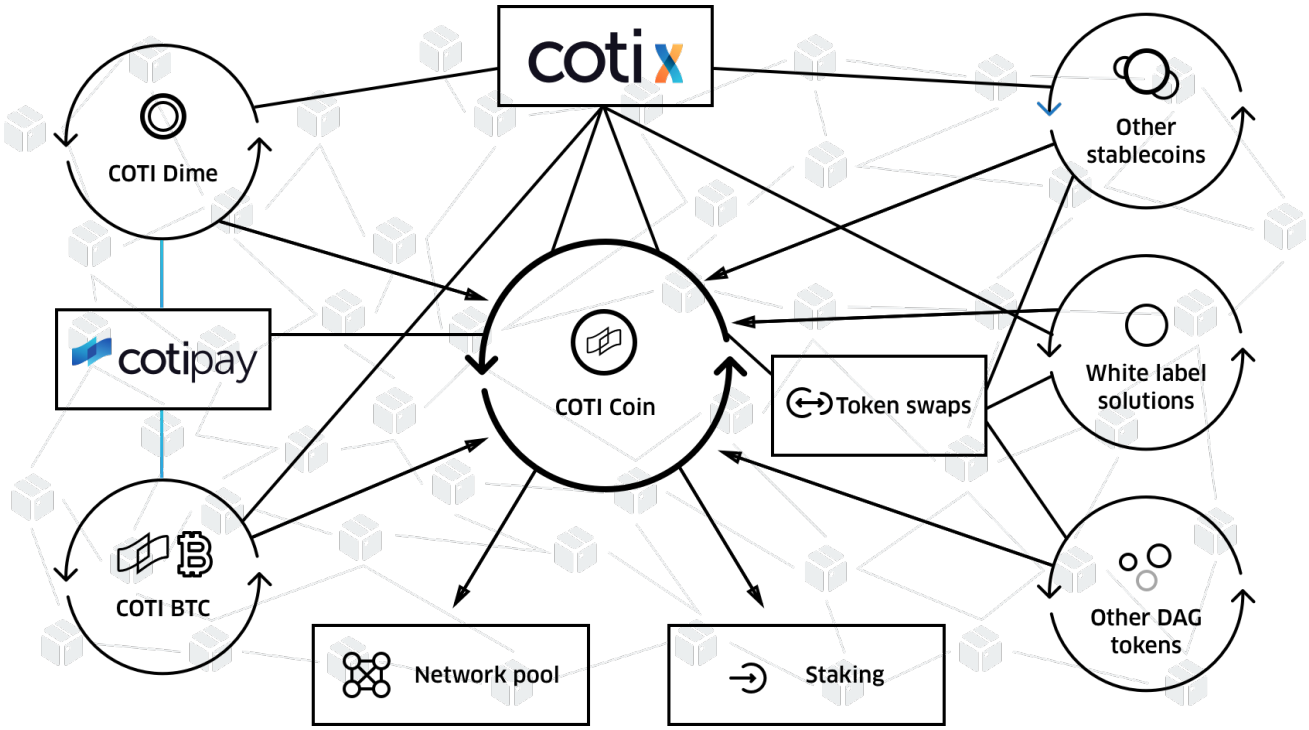
3.4 Dedicated coins

The COTI MultiDAG, another type of specialised coin, is designed for dedicated coins.

Dedicated coins are originated by enterprises that need their own currencies. Dedicated coins are useful for loyalty programs, discounts, coupons and more.

COTI dedicated coin mechanics are similar to the COTI stable coin, but are based on different rules set by the originator of the dedicated coin and implemented in its confirmation rules and smart contracts.

Figure 13: COTI ecosystem with coins of different types



4 Anatomy of a Payment

For simplicity, this section does not take the COTI MultiDAG structure into account.

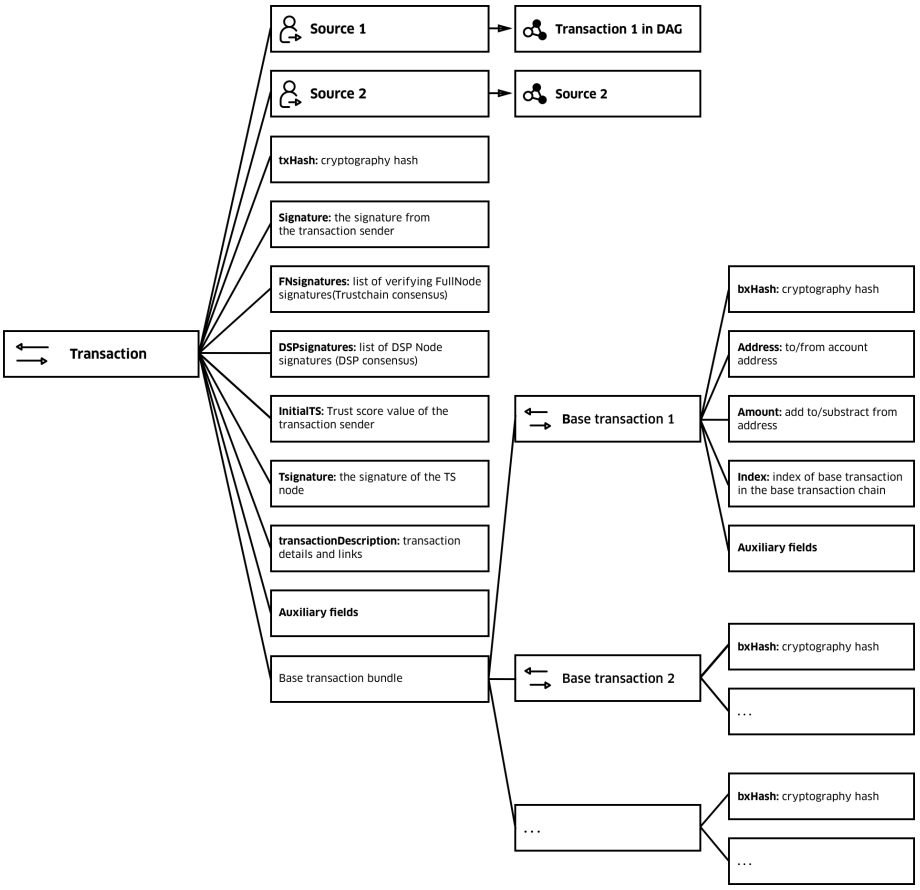
4.1 Accounts

In COTI, an account is a set of a user's cryptography addresses combined and managed together for the user's convenience. Addresses are cryptographically generated from the user's seed public keys. In COTI, users can choose how to use their addresses – as one-time or permanent addresses, or as addresses organised into several accounts with different rules. Addresses may not have a negative balance and are checked by Full Nodes and signed by DSP Nodes.

4.2 Transaction structure

A transaction in the COTI network includes descriptive fields, cryptography protocol related fields and a bundle of elementary base transactions.

Figure 14: The structure of a transaction



4.3 Fees

All the fees in the COTI network are collected by Full Nodes. Fee base transactions are created together with all base transactions in the bundle and are signed by the user and the transaction, creating a Full Node together with the whole transaction.

Network fee base transactions are also included in the main transaction bundle. These base transactions spend COTI from the Full Node addresses and are signed by the Full Node signature together with the whole transaction.

The network fee level is a network-wide constant at any given time.

The correctness of the network fee included in the transaction bundle is verified by other Full Nodes in the transaction confirmation.

4.4 Multisig accounts

COTI provides users with multisig accounts allowing escrow and other cryptography lock possibilities. Multiple signatures are checked by Full Nodes confirming transactions.

4.5 Rolling Reserve

The Rolling Reserve is a share of a seller's funds that is frozen for the purpose of buyer protection. Being a key element of the buyer protection system, the Rolling Reserve is crucial for online trade. In

COTI, the Rolling Reserve is implemented at the protocol level for maximum efficiency.

The Rolling Reserve account belongs to the COTI Arbitration system as a pseudo-user, but is linked to the Seller as the conditional owner of funds. Rolling Reserve account funds can be spent in two ways:

1. transferred to the seller's account at the end of the freezing term
2. used as a payment to the plaintiff if the seller loses a case.

The Rolling Reserve share and freezing term are dependent on the Seller's Trust Score the moment the payment transaction is created.

COTI is considering two possible options of Rolling Reserve handling. The first is having the payment split at the creation of the transaction bundle in the wallet (as seen in Figure 15). The second is having the payment transferred to the RR account in full and then split from the RR account (as seen in Figure 16).

Figure 15: Handling the Rolling Reserve by splitting the payment at the creation of the transaction bundle.

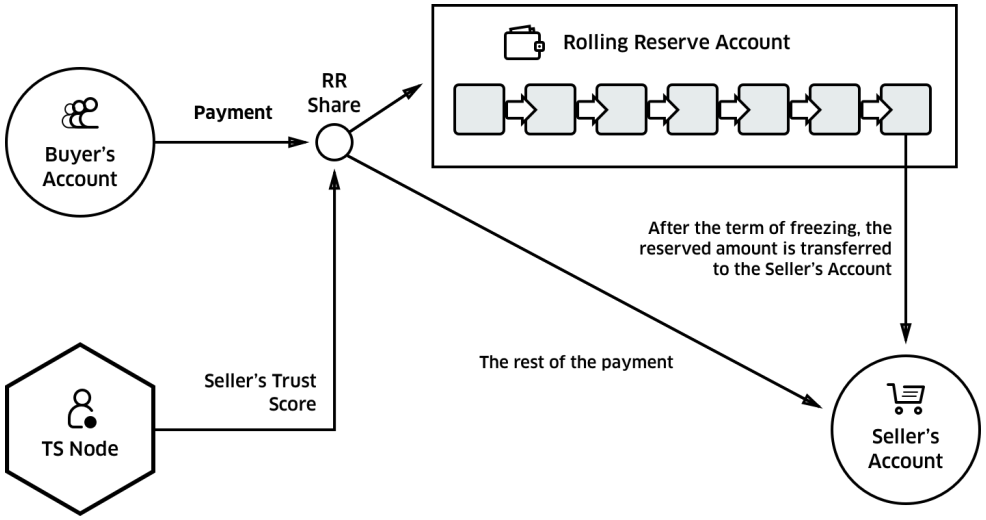
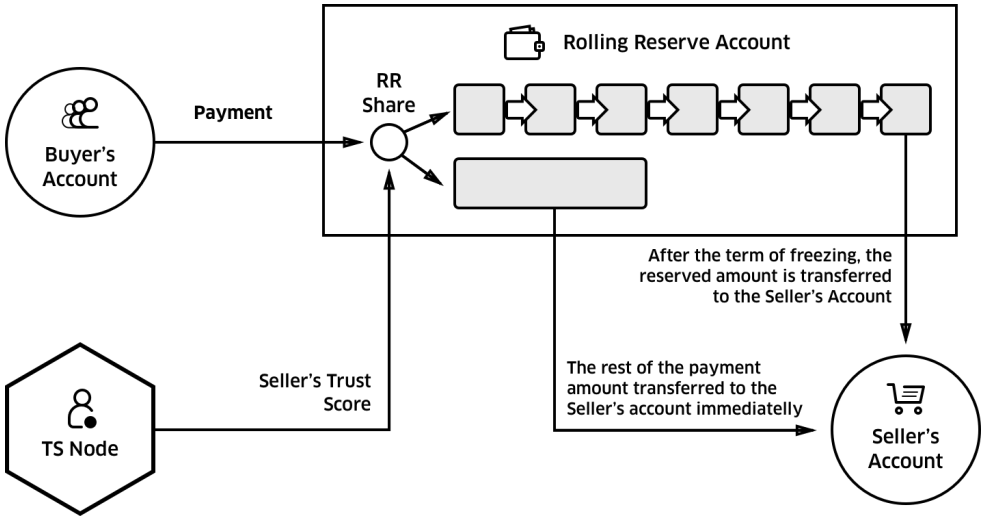


Figure 16: Handling the Rolling Reserve by having the payment transferred to the RR in full before splitting it.



5 Trust Score

Trust Scores are a key feature of the COTI network. They are used for effective transaction processing, risk mitigation and network structuring. Trust Scores are calculated by dedicated and decentralised Trust Score Nodes.

5.1 Trust Scores in common

Actors in the COTI network possess their own Trust Score metrics, including Nodes. A Node's Trust Score, however, is not the same as Node owner's Trust Score.

All accounts in the payment network have a Trust Score, which changes according to a user's payment statistics and in alignment with several event types. An account Trust Score is a real number within the range [0,100]. All transactions in the payment network also have a Trust Score. The Trust Score of a transaction is the sender's Trust Score when the transaction is initiated. The Cumulative Trust Score of transaction A is the sum of all the Trust Scores of all transactions along the *heaviest path* approving transaction A, including transaction A itself.

The Trust Score Algorithm is designed to ensure the maximal performance of the Trust Score-based Source Selection Algorithm.

For example, let the recommended level of trust (cumulative Trust Score) for the transaction be 1000. This means that transactions created by highly trusted network participants (TS $\sim$ 85) need at least 12 confirming transactions in the chain, while transactions from low trusted network participants (TS $\sim$ 12) need at least 84 confirming transactions in the chain.

Trust Score Nodes collect transaction and other statistics to continuously update participants' Trust Scores as described in the Trust Score Update Algorithm subsection below.

Trust Scores can also be changed according to the occurrence of external events, such as bankruptcy of the company that owns an account. The Trust Score Nodes will receive information on these events confirmed by Arbitrators.

5.2 Uses and Implications of the Trust Score

1. Arranging transactions for the Trustchain algorithm and providing optimal parallel transaction processing, while ensuring improved performance for highly trusted users and additional checks for less trusted users.
2. Defining the Rolling Reserve requirements for merchants.
3. Determining fee levels.
4. Setting PoW levels, which can indirectly affect fee levels.
5. Setting the Arbitrator threshold and defining which network participants can be chosen to be Arbitrators and be included in the Arbitrators Pool.
6. Defining network topology and transaction propagation paths (Node Trust Scores).
7. Optimising Node workload (Node Trust Scores).

5.3 Trust Score types

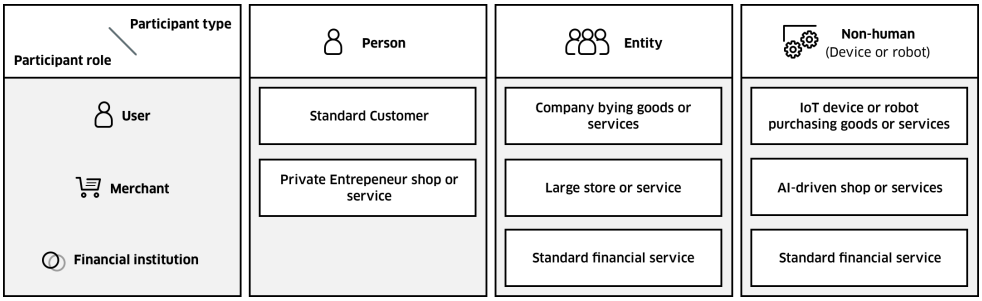
The various types of COTI network participants use different Trust Score counting algorithms.

A participant may be a person, a business entity or a nonhuman (e.g. the IoT device), and may have the role of a user, merchant or a financial institution.

The Node's Trust Score is described below in subsection 5.8, "Trust Score for Nodes."

When a participant is registered as a merchant or a financial institution, it enables the participant to receive payments in exchange for goods or services. Uncooperative participants attempting to sell goods or services without registering either as a merchant or a financial institution will be penalized with a low Trust Score (see the Trust Score range for malicious network participants below). As a result, such participants will be assigned high Rolling Reserve requirements amounting up to 100% in order to uphold buyer protection standards.

Figure 17: The different types of participants and roles in the COTI network.



5.4 Trust Score ranges

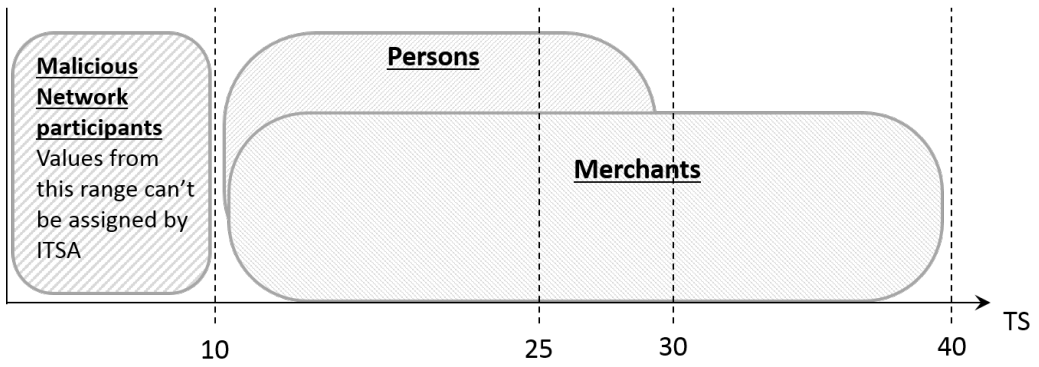
All Trust Scores are real numbers within the range [0,100].

A zero Trust Score means that the participant has been banned from using the COTI network for serious wrongdoing, including false identification or the submission of forged documents.

The Initial Trust Score Algorithm (ITSA) assigns TS values in the range [10, 30]. The [0, 10] range is reserved for malicious network participants. The initial TS values for legal entities and for all merchants lies within the broader range [10, 40] because these participants can supply more data related to their activities than a simple customer.

The Arbitrator Trust Score threshold will be preliminarily defined to 25, but this value will be adjusted according to actual network statistics.

Figure 18: The ranges of the Initial Trust Score for different user types.



5.5 Decentralised design of Trust Score Nodes

Trust Score Nodes are decentralised servers that provide the COTI network with the storage and computational power needed to calculate Trust Scores and supply network participants with them. Trust Score Nodes work using the same algorithm. Other Nodes control it by cross-checking Trust Score calculations.

5.6 Initial Trust Score Algorithm

The Initial Trust Score of a network participant is calculated using the Initial Transaction Score Algorithm (ITSA), based on machine learning (ML) and rules-based approaches, within COTI KYC Server.

The dataset that the ITSA uses to calculate the Initial Trust Score includes sensitive user details from the KYC/AML procedures and the questionnaire they were required to fill in. If users wants to supply additional details to increase their Trust Score, it is possible to fill in the questionnaire more than once.

5.6.1 Data used by the Initial Trust Score Algorithm for a user

It is not possible to join the payment network before completing KYC verification, so it defines the minimal amount of data the ITSA possesses and the minimal Trust Score a participant can have after joining the network. Identification document type, citizenship, date of birth, proof-of residence, phone number, country of residence and zip code are all KYC data types that are available to the ITSA as it runs within the COTI KYC Server.

The following data is requested from users filling in the Trust Score questionnaire: family status, education level, employment, occupation, income data, insurance, driver's license, bank reference, social network account name and web site, etc. It is up to User, to fill these data or not to fill, but usually the more data the User supplied, the large Trust Score he/she has.

If a person purports to have a higher Trust Score than the Arbitrator threshold, then the ITSA can randomly generate a request to submit supporting documents. The documents are then checked by a randomly chosen Arbitrator and sent to the Trust Score nodes. If the user fails to confirm the Trust Score questionnaire data, or if the documents are found to be forged, it will constitute a misdemeanour event and will be used to downgrade the user's Trust Score.

A proof of source of funds is required according to KYC/AML procedures in the event that a user plans to transfer large sums of money.

5.6.2 Data used by the Initial Trust Score Algorithm for legal entities

Legal entities are also required to complete KYC procedures prior to opening an account.

The data supplied by a legal entity includes incorporation data, owner data, beneficiary owners, directors, responsible officer identifications, shares, issued securities, balance sheets, auditor reports, and business profile data. There are a lot of elaborated methods to evaluate and rate a company, but these methods are different from country to country. COTI plans to implement Trust Scores for companies before the main net beta is launched.

5.6.3 Data used by the Initial Trust Score Algorithm for merchants

In order to participate as a merchant, the participant must provide the following information, in addition to the KYC documents and questionnaire: business activities, including the nature of business, historical performance data, licenses (if applicable), bank references and planned sales volume.

If a merchant is a company, the required data set is the same as for any legal entity.

5.6.4 Data used by the Initial Trust Score Algorithm for robots and IoT devices

COTI Trustchain platform allows creating efficient light-weight solutions usable for IoT devices and robots. To be participants of COTI network, anyone needs to have a TrustScore. The Initial Trust Score for devices and robots will be, in part, borrowed from the device owner, and, in part, calculated basing on the device safety and robustness to hacker attacks. COTI is considering to use HighIoT (high-iot.com) device security data to calculate IoT devices Trust Score.

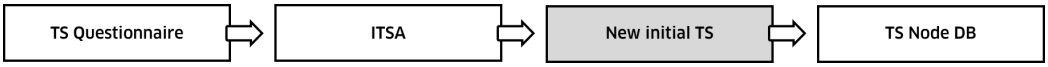
5.6.5 ITSA dataflow

Figure 19: The ITSA dataflow on submitting KYC data.

The minimal Trust Score range based only on KYC data is [10,15].

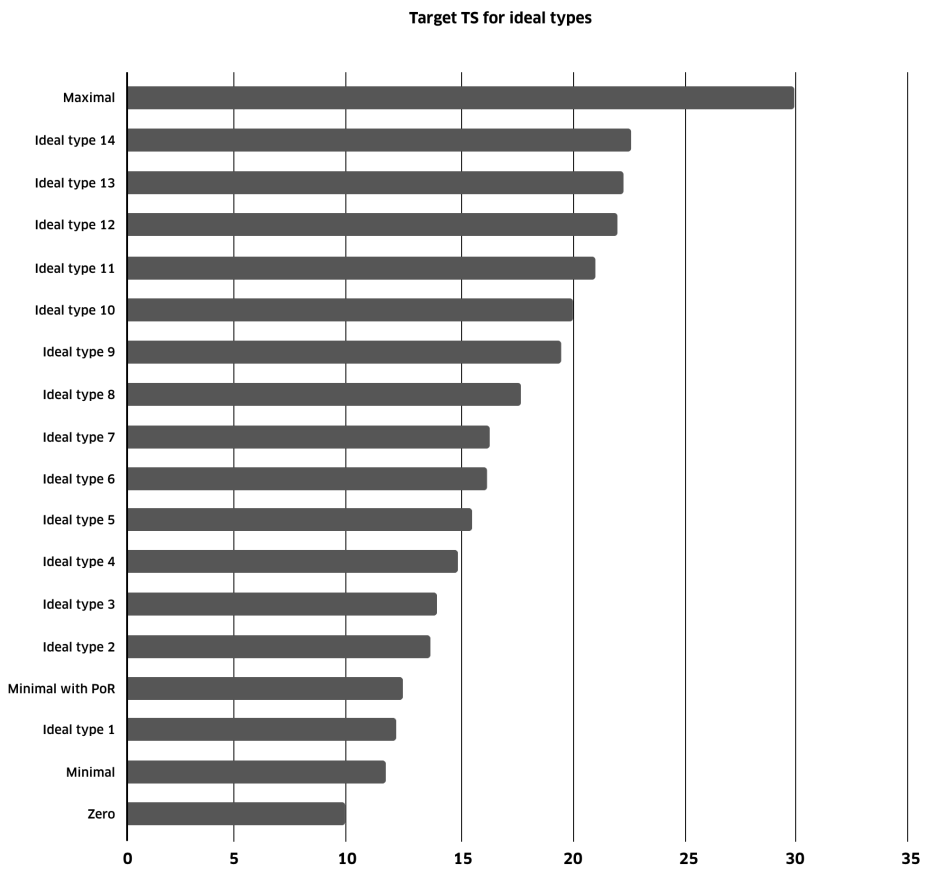
After beginning to use the network, participants can improve their Trust Score by filling in the Trust Score questionnaire and supplying additional data.

Figure 20: The ITSA dataflow on submitting the questionnaire supplying additional data.



The machine learning portion of the algorithm begins with the Ideal Types approach (see *M. Weber, The Objectivity of the Sociological and Social-Political Knowledge*). In order to have a labeled dataset, we defined 18 ideal user types (including 4 boundary points), together with desired Trust Score values for them.

Figure 21: TS for ideal types.



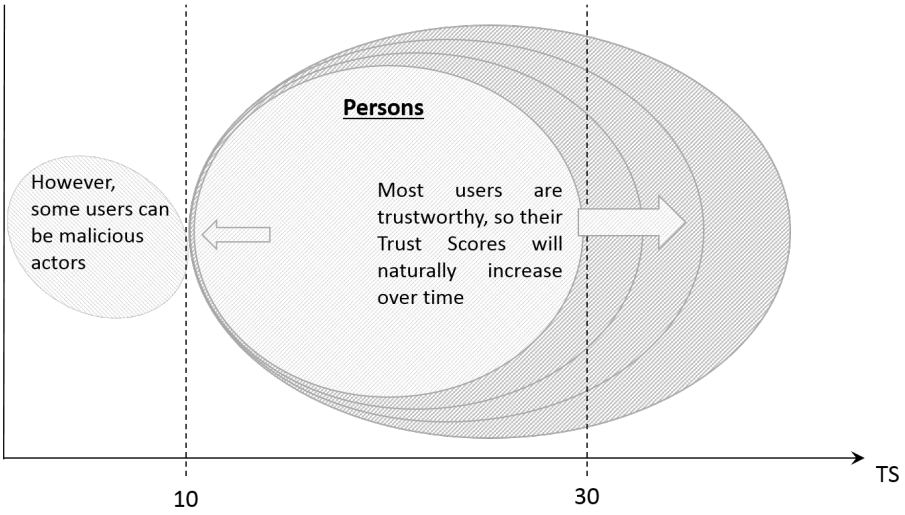
We took one hundred real people feature vectors and defined the desired Trust Score values according to the most similar ideal types. Having this dataset, we can use simple machine learning methods like linear regression to define the optimal set of weights used by ITSA.

The ITSA, however, doesn't only use a machine learning approach. For example, we have reliable statistics on cyber crimes by country and it is more reasonable to apply these statistics before using an ML algorithm, or to not even include such features at all. An individual's age is another parameter for which we have risk aversion statistics.

See ITSA code in COTI public github repository for further details.

5.7 Calculation of Trust Scores by the Trust Score Nodes

Figure 22: TS generally tends to increase from its initial value for most users.



5.7.1 Trust Score Update Algorithm

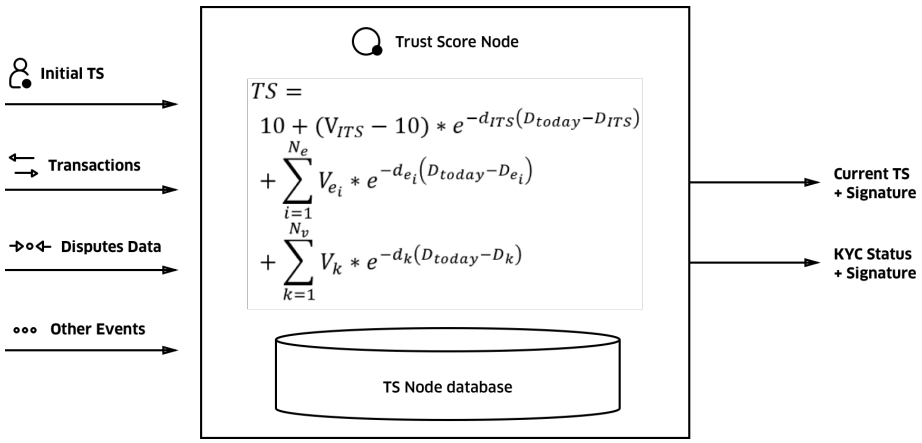
The Trust Score Update Algorithm (TSUA) has been designed to efficiently collect data on the user's behaviour in order to submit the information to decentralised Trust Score Nodes.

In the COTI network, Trust Score Nodes receive copies of all processed transactions and store them in their own database in shortened form to exclude the possibility of double counting and to enable Trust Score recalculations.

Trust Score Nodes maintain transaction, turnover and balance counters, while the TSUA recalculates them. They also manage Trust Score lists affecting events like arbitration outcomes, and decays for them.

There are two types of decay: exponential decay and limitation terms. Exponential decays are applied to numeric values, while limitation terms are applied to events. If there are no transactions or events involving a participant, then his/her Trust Score will converge asymptotically to 10.

Figure 23: The process of updating the TS.



5.7.2 Data for the Current Trust Score calculation

The current Trust Score value is calculated by a Trust Score Node using the datasets in the table (Table 1).

Table 1: Datasets used for the calculation of the Trust Score update along with their explanations.

Dataset	**Explanation**
(Initial Trust Score, date)	The initial TS assigned by the ITSA together with the corresponding date from which to calculate the decay.
Zero Trust Flag	An indicator for zero-trusted participants
Initial Trust Score Change Counters	Data compiled regarding the frequency of completing questionnaires and forging data that results in Trust Score downgrades.
[(Monthly turnover11, date), . . . (Monthly turnover0, date)]	The list of monthly turnovers for the previous 12 months, including the current turnover and dates for decay calculation.
[(Monthly average balance11, date), . . . (Monthly average balance0, date)]	The list of monthly average balances for the previous 12 months, including the current monthly average and the dates for decay calculation.
[(Monthly counter11, date), . . . (Monthly counter1, date), (Monthly counter0, date)]	The list of monthly transaction frequencies for the previous 12 months, including the current frequency and the dates for decay calculation
[(Misbehavior case1, date), . . . (Misbehavior caseN, date)]	The list of all known wrongdoing events and dates for decay calculation
[(Dispute outcome1, fulfilment, date), . . . (Dispute outcomeN, fulfilment, date)]	The list of all disputes and information on fulfilment and dates

5.7.3 Current Trust Score calculation

Event contributions are limited to T_n terms and are subject to exponential decay. In (5.1), T_1, T_1, ... T_N are periods of time; e_i refers to the i^{th} event and D_{e_i} and f_{e_i} refer to the date and contribution corresponding to this event.

$$V_{e_i} = \begin{cases} f_{e_i,1} & \text{if } D_{today} - D_{e_i} < T_1 \\ f_{e_i,2} & \text{if } T_1 \leq D_{today} - D_{e_i} < T_2 \\ ... & \\ 0 & \text{if } T_N \leq D_{today} - D_{e_i} \end{cases} \tag{5.1}$$

The contributions from additional numerical data ($V_k = f_k(v_k)$) – these may come from the Initial Trust Score (for example) – are calculated before applying the exponential decays. The current Trust Score is then calculated as follows:

$$TS = 10 + (V_{ITS} - 10) \cdot e^{-d_{ITS}(D_{today} - D_{ITS})} + \sum_{i=1}^{N_e} V_{e_i} \cdot e^{-d_{e_i}(D_{today} - D_{e_i})} + \sum_{k=1}^{N_v} V_{e_k} \cdot e^{-d_k(D_{today} - D_k)} \tag{5.2}$$

In (5.2) d_{e_i} and d_k are exponential decay factors, D_{e_i} and D_k are dates for events and numerical values, V_{ITS} is the Initial Trust Score value.

If the **Zero Trust Flag** is true, then TS = 0 without further calculations. If the calculated TS value is zero or negative, the participant's TS is set to be the minimum positive TS value (0.1): no one can be banned from using the network on the basis of a calculation. If the calculated TS value is greater than the maximum TS value, the participant's TS is set to be the maximum TS value (100). The calculated TS value is signed by the TS Node.

See TSUA code in COTI public github repository for details.

5.8 Trust Scores for Nodes

In the COTI network, Trust Scores are a universal concept that are applicable not only to users, but also Nodes. During Node creation, the Initial Trust Score value is set according to the owner's Trust Score (with a scaling coefficient)[6]. After initialisation, these Trust Scores become independent of one another.

Node Trust Scores are computed in a similar way to users, but with different parameters. The primary set of parameters consists of transaction generation frequency, transaction propagation flow, transaction confirmation frequency and other load parameters. The more work a Node does for the COTI network, the greater its Trust Score.

COTI needs to take into account value flow related characteristics, including Node centrality measures in order to calculate Trust Scores.

Node Trust Scores are further affected by negative events like issuing invalid transactions, double spending attempts, incorrect transaction attachment algorithm uses, evading to participate in smart contracts execution, etc.

Current Node Trust Score calculations are carried out similarly to that of users (see section 5.7.3), but utilising various contribution functions, weights and decays.

Node Trust Scores are updated using copies of all processed transactions received by the Trust Score Nodes simultaneously with the participant's Trust Scores.

[6]The assumption here is that the owner of a node is already a COTI user. If this is not the case, a standard value may be used.

Node Trust Scores are used to define COTI network topology and to help users select the most suitable Full Node. Low trusted DSP, TS or History Nodes are banned from network operations and do not receive any fees.

6 Network Components

6.1 Wallets

All clients in the system use a COTI wallet to manage their accounts, transfer funds to different addresses and check their current balance. The wallet contains a seed that is used to generate private and public keys (addresses). The seed is generated privately from the user's secret key and the user's server key generated by the KYC server. If lost, the seed can be recovered from the user's secret key and server key. In order to receive COTI network data and initiate transactions, a wallet connects to the Full Node chosen by the user. If the user has no preferred Full Nodes, a Node will be selected randomly. COTI's wallets are customisable to ensure that a user's wallet is to his or her satisfaction. As there are several types of participants and roles a user can register for (see Figure 17), the detailed operations of these wallets will vary from user to user.

6.2 The Nodes

COTI provides a decentralised solution designed to enable secure and trustworthy payments. This solution relies on the distribution of Cluster responsibility to several types of Nodes, which are run by users. There are four node types in the COTI network: Full Node, DSP Node, Trust Score Node, and History Node.

Full Nodes: are the main client facing servers of the system. Each wallet is connected to a Full Node and every transaction is received by a Full Node and propagated to the entire system. Together with DSP Nodes Full Nodes are the backbone of the system. Full Nodes are responsible for the Trustchain Consensus. They receive new transactions from the Wallets, validate them, do PoW and attach them to the DAG. Full Nodes also execute COTI smart contracts in a decentralized manner. In the COTI network, Full Nodes can define their own price list for users and compete for users. Consequently, COTI Full Nodes are responsible for collecting all fees for all protocol usage and transferring the Network Fee to the Network Pool. Please refer to the COTI Node Model Business Plan for further details. According to projections, COTI Full Nodes are expected to be profitable[7]

The Double Spending Prevention Nodes (DSP Nodes) are the set of highly trusted distributed servers responsible for DSP Consensus, account balance control and general protocol and data integrity. Each transaction has to be approved and signed by the majority of the DSP nodes. Users cannot directly connect to DSP Nodes.To run a DSP Node, a user is required to deposit a substantial amount of COTI. This amount can be deposited by the DSP Node operator alone or by a group of network participants delegating their deposits to the node operator. Please refer to the Double spend Prevention and DSP Consensus section for more details.

Trust Score Nodes are dedicated servers for calculating and storing Participant Trust Scores and Participant KYC statuses. To run a Trust Score Node, a user is required to deposit a sum of COTI. Please refer to the Trust Score section for more details.

The History Nodes keep the earlier parts of the Cluster after the Clusterstamp process is complete. Full account history can be retrieved from the History Nodes.

DSP Nodes, Trust Score Nodes and History Nodes are also paid from the Network pool. According to estimations, COTI Nodes are expected to be profitable[8].

[7]Please refer to the COTI Node Model Business Plan for further details.

[8]Please refer to COTI Nodes Model Business Plans Book for details.

6.3 Servers

COTI will maintain servers that will help the network to run smoothly. Two types of such servers are described below.

KYC Server: the COTI network's onboarding area, where new users connect for KYC/AML verification purposes. KYC Servers are also responsible for calculating a user's Initial Trust Score and creating server keys for user seed generation. The server keys created are stored on the KYC server and can be used during the seed recovery process. A user is required to connect to a COTI KYC server at least once for KYC verification purposes. Following the first KYC onboarding, users will not need to connect to it again unless they would like to update their Trust Score questionnaire. Please refer to Trust Score section for more details on users' Initial Trust Score.

Zero Spend Servers: responsible for sending zero-value transactions when a source in the Cluster has waited a long time without being validated by another transaction, or if a transaction cannot attach to a source using the Source Selection Algorithm. The activity of these servers will help to monitor the Source Selection Algorithm. If these servers experience over-activity, it may indicate that there are problems with the network or with the Trust Score Algorithm.

7 Proof of Work

In COTI, PoW is not as important as it is for Bitcoin, as it is only used to protect the network from spamming attempts. Nevertheless, all Full Nodes perform PoW when attaching new transactions. As such, COTI is not based on PoS, but is rather a highly scalable PoW cryptocurrency.

Proof-of-work (PoW) schemes are designed to be difficult to solve, but relatively easy to verify. Unfortunately, most PoW systems achieve fast verification simply by verifying one round of parallel search algorithms. These parallel algorithms are quickly adapted to graphics cards, FPGAs, or even ASIC designs, which would give an attacker an advantage by several orders of magnitude over the common computer. In the case of cryptocurrencies, where the primary goal of PoW is trust decentralisation and participant incentivisation, it becomes critical that PoW not be optimised and accelerated by FPGA or ASIC designs with any meaningful economic return on investment. One approach that has been adopted by many cryptocurrencies is to use what is known as a Sequential Memory-Hard Function, such as Scrypt. These algorithms attempt to make the PoW dependent upon sequential random access to a large array of data and thus be memory constrained to limit parallelisation. The challenge with sequential memory-hard functions is that when they are tuned to use large amounts of memory, they lose the property of being easy to verify. For example, simply populating 1 gigabyte of memory with cryptographically secure pseudo-random data can take a second to perform. As a result, the requirement to validate such a memory-hard PoW would create an opportunity to perform a denial of service attack. Therefore, we see a need for a family of memory-hard PoW algorithms that can be validated in milliseconds while requiring a lot of memory to efficiently find a solution.

7.1 Memory hard PoW

The main reason why memory hardness is important is to make the PoW function resistant to specialised hardware. Bitcoin, with a mining algorithm that only requires a simple SHA256 computation, has led to the creation of companies that specialise in the manufacture of application-specific integrated circuits (ASICs) for the sole purpose of computing billions of SHA256 hashes to mine Bitcoin blocks. These chips have no legitimate applications outside of Bitcoin mining and password cracking, and the presence of these chips, which are thousands of times more efficient per dollar and kilowatt hour at computing hashes than generic CPUs, makes it impossible for ordinary users with generic CPU and GPU hardware to compete.

This dominance of specialised hardware has several detrimental effects:

- It negates the democratic distribution aspect of cryptocurrency. In a generic hardware-dominated ecosystem, the fact that everyone has a computer guarantees that everyone will have an equal opportunity to earn at least some of the initial money supply. With specialised hardware, this factor does not exist; each actor's mining potential is linear (in fact, slightly superlinear) in their quantity of pre-existing capital, potentially exacerbating existing wealth inequalities.|
- It increases resource waste. In an efficient market, marginal revenue approaches marginal cost. Since mining revenue is characterised by a wide range due to money spent on mining hardware and electricity, this also implies that total revenue approaches total cost. Hence, in a specialised hardware dominated ecosystem, the share of resource waste is close to 100% of all network consumption.

  Because everyone already has a computer in a CPU and GPU-dominated ecosystem, people do not need to buy specialised hardware for the first few hashes per second worth of mining power. Hence, revenue is sublinear in cost-everyone gets a certain amount of revenue for free. This implies that the quantity of resources wasted by the network is potentially lower than its security parameter.
- It centralises the network to a few actors (ie. ASIC manufacturers) This makes 51% attacks much more likely and potentially opens the network to regulatory pressure.

Specialised hardware manufacturers can certainly pack terabytes of memory into their devices, but this effect is mitigated by two factors. First, hobbyists can achieve the same effect by simply buying many off-the-shelf memory cards. Second, memory is much more expensive to produce (if measured in laptop equivalents) than SHA256 hashing chips, while the RAM used in ordinary computers is already optimised. To achieve the goal of being trivial to verify but memory intensive to solve, the PoW must be asymmetrical in terms of the amount of memory required to validate the work. As a consequence, the individual steps of the PoW must be parallel because they are the foundation of the validation step. Despite parallel steps that can be run in less than a millisecond, algorithms can be made memory-hard by requiring a solution that depends upon the relationship between two or more of the parallel steps, thereby benefiting from the storage of every parallel step's result. The results can be quickly verified by performing just two or three parallel steps and checking the relationship between the outcomes.

7.2 How is the PoW in COTI better?

COTI uses PoW for spam protection and network participant incentivisation (node operators), similar to the PoW used in Hashcash. This is a short computational operation, which should not be confused with the expensive PoW employed in miner-based ledgers like Bitcoin. The PoW in the COTI protocol isn't better or worse than the PoW in Bitcoin or any other mining PoW, but serves different purposes. In the Bitcoin and Ethereum protocol, doing PoW is a way to define the truth. It means that if you resolve a block faster than anyone else, that block will be the truth immediately after being validated by peers. In COTI, PoW is just a way to prevent spamming and also to balance incentives for network participants. Finding a suitable nonce (i.e. doing your PoW) allows you to attach your transaction to the DAG, but doesn't decide the truth. With this key difference in mind, it's obvious that the Bitcoin/Ethereum PoW difficulty must be far greater than the PoW in COTI because it carries far greater power. The PoW in Bitcoin and Ethereum is a central point to consensus. In COTI, the PoW is just a protection mechanism with almost no impact on consensus, except that transactions with higher difficulty are handled by Full Nodes.

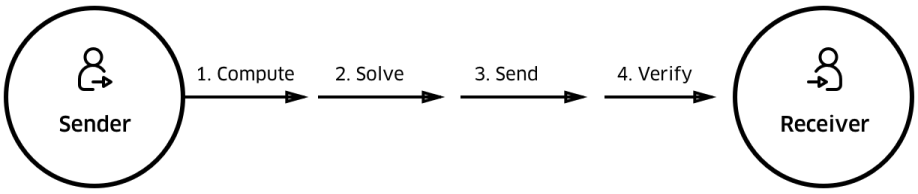
7.3 COTI (AlphaNet) PoW

The history of hashing for cryptocurrencies began with SHA256 for Bitcoin, then Scrypt for Litecoin, Ethash for Ethereum and X11 for Dash, followed by X13, X15, and X17. The reason for the algorithm changes is to minimise the impact of purpose-built hardware on the mining ecosystem of the coin. Bitcoin was originally intended to be mined by computers everywhere. As the value of Bitcoin increased, it

became advantageous to mine using hardware designed for parallel processing, so the mining moved to Graphics Processing Units (GPUs). As the economic value of mining further increased, it became economically viable to use programmable hardware in the form of Field-programmable Gate Arrays (FPGAs), which had an advantage over CPUs and GPUs. The next step was to build custom chips purpose-built for mining. These Application Specific Integrated Circuits (ASICs) were able to dominate competing technologies and made it impractical to mine any other way. The last, and likely final iteration for Bitcoin mining, is the move to faster and more energy efficient ASIC hardware. The unfortunate side effect of this transition to ASIC hardware is mining centralisation. While anyone can order these ASICs, there is an advantage to being near the manufacturing facility as shipping time is reduced. Additionally, access to cheap electricity is a priority, as the electricity used is the variable cost of the mining operation. This has led to some centralisation of mining in China because of the proximity to ASIC development and the availability of inexpensive electricity in some provinces. One solution to minimise the impact of ASIC miners is to use a memory intensive hashing algorithm. This is the approach of Scrypt, used by Litecoin, and Equihash, used by ZCash. These two algorithms have reduced the impact of ASICs. While there are some ASIC miners for Scrypt, the relative advantage over GPUs is negligible. There are currently no ASIC miners for Equihash.

Another approach is to use a sequence of hashing algorithms where the output of one becomes the input of the next. Dash, formerly DarkCoin, took this approach with their X11 algorithm. X11 uses eleven chained hashing algorithms in an effort to thwart the 1 move to ASIC mining. This approach worked for a while, but several manufacturers now produce ASIC miners for X11. The concept behind X11 can be extended to additional algorithms. For this reason, some coins use X13, some X15, and even X17, which chains seventeen hashing algorithms. The fixed order of hashing algorithms lends itself to the construction of ASICs. While chaining more algorithms together adds difficulty in constructing an ASIC, the X13, X15, and X17 all use the same ordering of hashing algorithms as the X11. This is likely to lead to faster manufacturing of ASICs for these algorithms as manufacturers only need to extend their existing design to accommodate the additional hashing algorithms.

Figure 24: The Proof of Work interaction.



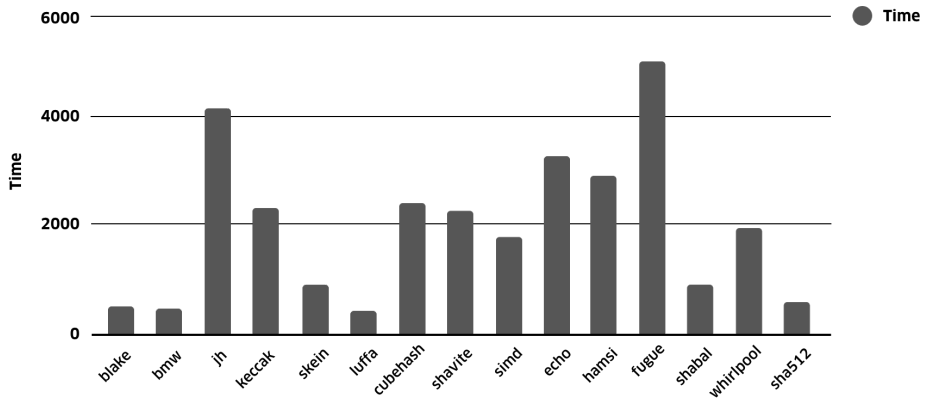
The COTI PoW algorithm intends to solve this problem by constantly disrupting the ordering of the hashing algorithms. The hashing algorithms are the same proven algorithms used in X15 + SHA512, but the ordering is changed based on the hash of the previous block. This reordering does not make an ASIC impossible to build, but it does require that the ASIC adapt to additional input, which is more easily accomplished by a CPU or GPU. The reordering also prevents a simple extension of the current X11 ASICs or future X15 ASICs. The COTI PoW hashing algorithm consists of multiple hashing algorithms operating in chain fashion with the ordering dependent on the last 16 bytes of the hash of the previous block.

Table 2: The basic algorithms used in COTI's PoW Algorithm.

1=blake	A=echo
2=bmw	B=hamsi
3=groestl	C=fugue
4=jh	D=shabal
5=keccak	E=whirlpool
6=skein	F=sha512
7=luffa	
8=cubehash	
9=shavite	

Example: Block Hash: 0000000000000000007e8a29f052ac2870045ae3970270f97da00919b8e86287 The final 16 bytes: 0x000000007da11919b8e86287 Each hex digit (nibble) determines which algorithm to use next. cubehash → shabal → echo → blake → blake → simd → bmw → simd → hamsi → shavite → whirlpool → shavite → luffa → groestl → shavite → cubehash

Figure 25: Relative Time per Hash Algorithm.



Some of the hash algorithms take longer than others. This time differential tends to average out across the algorithms and can be used to adjust the time for receiving a semi constant average execution. The concepts behind COTI PoW could be extended to include Scrypt, Equihash, and other ASIC resistant algorithms. The ordering of the algorithms can easily be changed for each infrastructure in order to dissuade hardware manufacturers from building ASICs for an entire class of coins as with X11. As such, COTI PoW should take the time of each algorithm into account and assign a concatenation of hash algorithm to process based on the Trust Score (TS) + Difficulty Level (DL) + Amount normalisation range. Possible ranges based on TS, for example, are as in the following:

Table 3: The various PoW algorithms used for different ranges of Trust Score.

Trust Score	**Algorithm Used**	**Average Time For PoW**
0-10	hamsi → fugue → groestl → simd → echo → luffa → cubehash → whirlpool → jh → shavite → blake → skein → sha512	30 sec

Table 3 – continued from previous page

Trust Score	**Algorithm Used**	**Average Time For PoW**
10-20	fugue → groestl → simd → echo → luffa → cubehash → whirlpool → jh → shavite → blake → skein →sha512	24 sec
20-30	groestl → simd → echo → luffa → cubehash → whirlpool → jh → shavite → blake → skein → sha512	16 sec
30-40	simd → echo → luffa → cubehash → whirlpool → jh → shavite → blake → skein → sha512	12 sec
40-50	luffa → cubehash → whirlpool → jh → shavite → blake→ skein → sha512	8 sec
50-60	cubehash → whirlpool → jh → shavite → blake → skein → sha512	5sec
60-70	whirlpool → jh → shavite → blake → skein → sha512	4 sec
70-80	shavite → blake → skein → sha512	3 sec
80-90	blake → skein → sha512	2 sec
90-100	skein → blake	1 sec

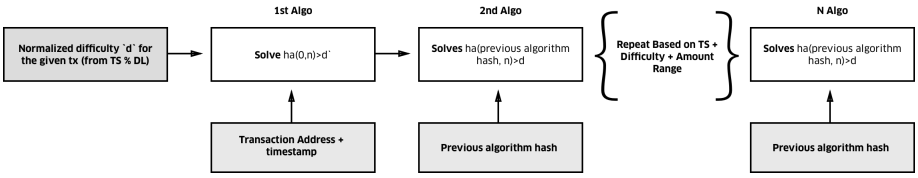
7.4 The PoW Algorithm in a Nutshell

1. Based on the TS + DL + Amount, the values should be normalised in the range of 0-100.
2. From the algorithm group set we should select and generate the hash of all algorithms sequentially concatenated.
3. Each algorithm in the group set should be selected only once until all are used and the time threshold is reached.
4. The execution of each hash is done sequentially, meaning the next hash is dependent on the result of the previous hash. In this way the possibility of execution parallelism is blocked while minimising execution time.
5. The idea is to start with the first algorithm and use each algorithm that needs to find its own nonce to solve for the current normalised difficulty (some mathematical combination of TS and DL). The nonce along with the valid hash would be used as input for the next algorithm. In this way, a set of nonces can be generated and used as input, in addition to the previous algorithm's hash, in order to verify a complete PoW cycle.

Algorithm 3: Pseudocode for solving COTI's combined hash PoW algorithm.

```
Given:
  d // normalised difficulty for the given txn (from TS & DL)
  deterministic_order // from the hash of the previous block
for each ha in deterministic_order: // ha a hashing algorithm
  if no previous algorithm:
      previous algorithm hash = 0
  find n such that ha(n, previous algorithm hash) > d
  save n
```

Figure 26: Proof of Work flow.



6. The next phase is to take the previous generated hash and use it as the input/public key for the next algorithm.

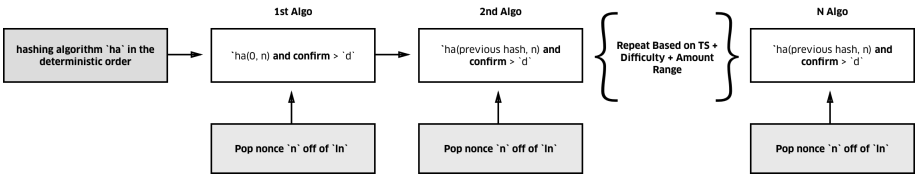
7. PoW validation should be instant. The validation process should be able to validate the proof of effort, meaning that all algorithms were indeed executed and performed properly. It could be the same nonce with a concatenation of each algorithm's result.

Algorithm 4: Pseudocode for the validation stage of COTI's PoW.

```
Given:
  ln // a list of nonces
  d // the normalised difficulty for the given transaction
    deterministic_order // from the given txn
for each ha in deterministic_order: // ha a hashing algorithm
  pop n of ln // n a nonce
  compute k = ha(n, previous hash)
    confirm k > d
```

Validation works on each block by validating the algorithm signature with the public key provided. In the case of cycles [1-N] the validation is run against private and public keys.

Figure 27: Proof of Work validation flow.



7.5 Normalising TS + DL

Our scaling will need to take into account the possible range of the original Trust Score and difficulty numbers. So let:

- denote the minimum of the range of TS + DL
- denote the maximum of the range of TS + DL
- denote the minimum of our desired target scaling - 0
- denote the maximum of our desired target scaling - 100 $m \in [r_{min}, r_{max}]$
- denote our measurement to be scaled

Then

$$m \to \frac{m - r_{min}}{r_{max} - r_{min}}(t_{max} - t_{min}) + t_{min}$$

will scale m linearly into $[t_{min}, t_{max}]$ as desired. To go step by step,

1. $m \to m - r_{min}$ maps m to $[0, r_{max} - r_{min}]$
2. Next , $m \to \frac{m - r_{min}}{r_{max} - r_{min}}$ maps m into $[0, 100]$

3. Multiplying this by $t_{max} - t_{min}$ maps m to $[0, t_{max} - t_{min}]$.
4. Finally adding t_{min} maps m to $[t_{min}, t_{max}]$.

7.6 Scaling Difficulty

For cryptocurrencies, it is not enough that PoW be memory hard, it must also be flexible enough to scale the difficulty of the work to finely tune the block production rate. For this reason, the final step of COTI PoW is to also adjust the work based on the difficulty level defined by the network.

8 Double spend prevention and DSP Consensus

8.1 DSP Consensus

For all high-performance distributed ledgers, potential double spending attacks are a fundamental problem. High performance is achievable only with a high degree of parallelism, and the price for this is a non-coherent state of network portions. Most known solutions to the problem are inefficient or centralised. The COTI double spend prevention solution consists of adding a handful of highly trusted Nodes to the network with only one function: to reach consensus whether the transaction is legitimate or a double spend. DSP Consensus consists of a majority of DSP nodes. When a transaction has more than one half of the DSP Node signatures, then consensus is achieved.

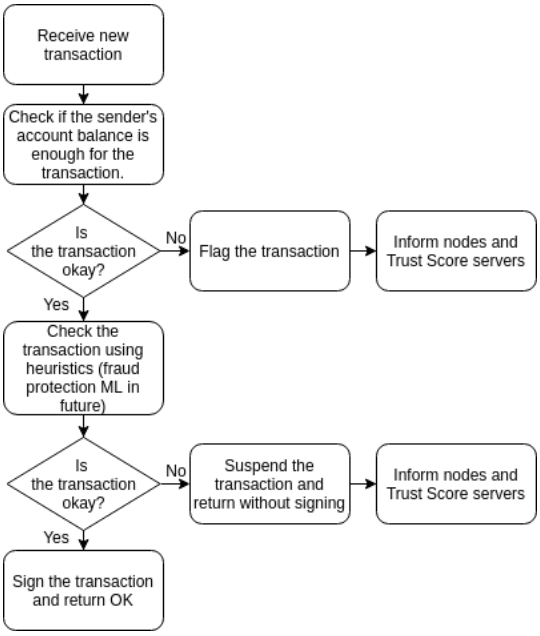
8.2 The Double-spend Prevention Mechanism

As shown in Figure 28, in order to do prevent double-spending, the DSP Nodes:

1. Keep a light version of the Cluster with pre-calculated balances for all accounts;
2. Receive a copy of any new transaction attached to the Cluster;
3. Check new transactions against a set of heuristics to detect possible double-spending attempts;
4. Check new transactions against available account balances;
5. Sign legitimate transactions;
6. Flag transactions suspected of double-spending;
7. Inform Trust Score Servers about double-spending attempts.

As the transaction verification process performed by the DSP Node is supposed to be a quick operation, only the amounts involved are checked, as opposed to the signatures of a transaction. The checks that the DSP Nodes perform are only carried out after a transaction has been attached to the Cluster. Transactions require the signature of a DSP Node before they can be considered fully confirmed. Any double-spending attempts detected are flagged and refused if malicious, while valid transactions receive signatures from DSP Nodes. Any valid transaction should receive a number of signatures defined by the consensus in order to continue as confirmed. The DSP Nodes are load balanced to ensure that the verifications that prevent double-spending are fast.

Figure 28: The verification procedure followed by DSP Nodes

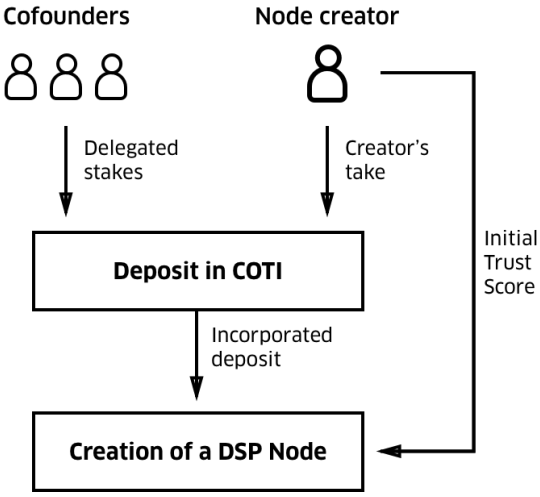


8.3 The creation of a DSP Node

Due to the nature of the verifications required to prevent double-spending, a user who would like to run a DSP Node has to meet the following requirements:

1. A user should have a high enough Trust Score in order to serve as a DSP Node operator;
2. A substantial amount of COTI will have to be deposited in a special multisig account;
3. The performance and security of the DSP Node must be checked remotely, including the quality of load balancing.

Figure 29: Procedure for DSP Node creation based on the principle of delegated proof-of-stake.



9 The Clusterstamp

To prevent the growing Cluster from becoming unmanageable in storage size, COTI has implemented the Clusterstamp, which consists of two phases:

1. The last fully confirmed transactions (having both Trustchain consensus and DSP consensus) for each account have been found. The 'last' transaction means that there are no fully confirmed transactions confirming it. These transactions then become the genesis transactions in the next generation of the Cluster.

2. All other confirmed transactions are excluded from the working DAG kept by Full Nodes. All non-confirmed transactions are kept in the working DAG.

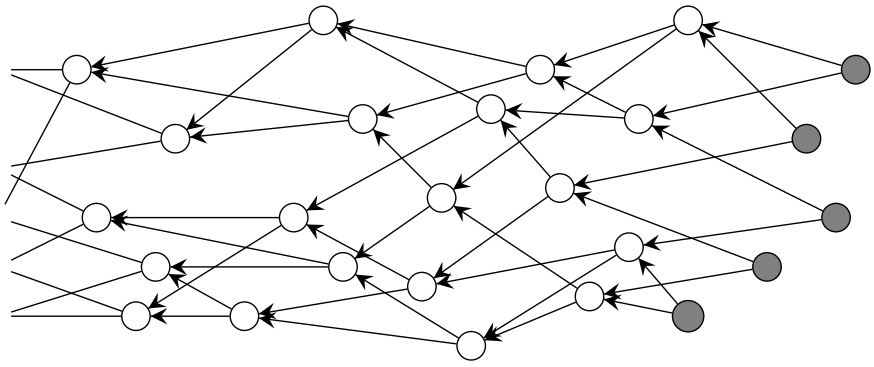
The Clusterstamp process is performed automatically. Following the creation and verification of a Clusterstamp, it is stored in the COTI History Nodes.

History nodes are receiving copies of all propagated transactions together with confirmation states, for this reason we don't need to copy any transaction data during the Clusterstamp.

The Clusterstamp is not applicable to the smart contracts Cluster.

Besides keeping the DAG operational as a data structure, the Clusterstamp has more benefits for the COTI network. The Clusterstamp provides a useful reference point and an opportunity for performing a system-wide audit to ensure that there are no inconsistencies or possible fraud.

Figure 30: The Clusterstamp process captures all the information up to the time of the gray transactions. Thereafter, the next generation of the Cluster begins with the gray transactions.



10 Performance Investigation

In order to better understand the performance characteristics of COTI's algorithms, we will provide a mathematical framework for making deductions about the Cluster, in addition to high level mathematical observations in a simplified context. We will then present a series of empirical investigations that make use of a full simulation of the Cluster.

10.1 Mathematical framework

For the purposes of analysis, we have made some simplified assumptions about the Cluster and the transactions taking place within it. First, we assume that all nodes take a fixed amount of time Δt to run the Source Selection Algorithm and perform proof-of-work. Second, we assume that new transactions arrive according to a Poisson process with fixed rate λ. We also assume that the transactions are all valid and the senders are distinct. All of these assumptions are not reflective of the real world, but can be locally true for stretches of time and sections of the Cluster, and are therefore useful to consider for analysis. The parameters λ and Δt will feature in the discussions below.

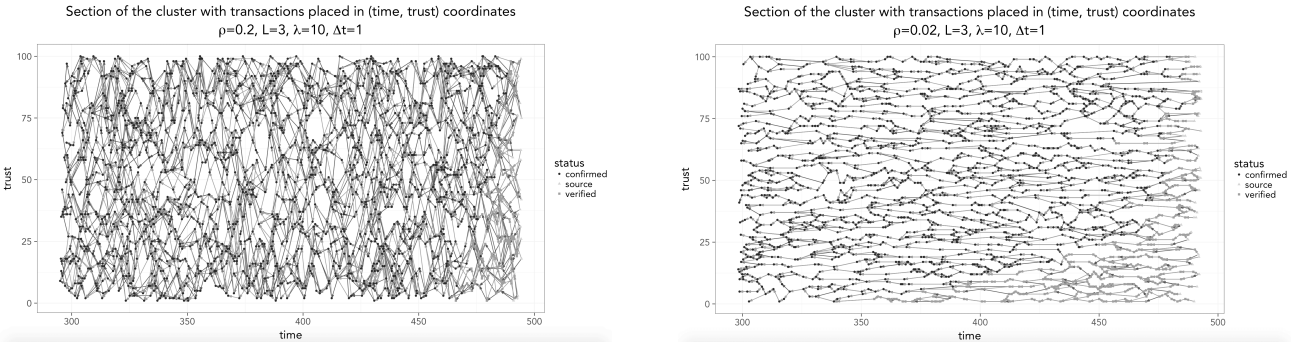
There are also a number of internal parameters that control how the Cluster will operate. Trust Scores can take on integer values from 1 to d, the maximum possible Trust Score. We will assume $d = 100$ in what follows. When a new transaction arrives in the Cluster, we assume it has Trust Score i, $(1 \leq i \leq d)$ with probability $P_{TS}(i)$, where $P_{TS}(1) + P_{TS}(2) + ... + P_{TS}(d) = 1$. This corresponds to the assumption that transactions of each Trust Score arrive with independent Poisson processes, wherein

the process for Trust Score i has rate $\lambda P_{TS}(i)$. An additional internal parameter L appears in the Trust Chain Algorithm. This parameter determines if a transaction is confirmed when the highest trust path from that transaction to a source transaction in the Cluster has a cumulative Trust Score of at least Ld.

Finally, the Source Selection Algorithm enables a new transaction to attach to any source with a sufficiently similar Trust Score and is controlled by a parameter ρ. If the set of available sources has size S and is sorted according to differences in Trust Scores from the new transaction, the first ρS must be available for selection, along with any others with the same Trust Score. The Source Selection Algorithm also depends on another parameter, R, which dictates the maximum absolute Trust Score difference allowed between a transaction and the transactions it approves. For the sake of analysis, we assume that $R = d = 100$.

The set of parameters $(\lambda, \Delta t, d, L, \rho, P_{TS}(\cdot))$ fully determines this simplified model of the Cluster. It is useful to visualise the transactions of the Cluster in the 2-D space described by time on the x axis and Trust Score on the y axis. The DAG structure formed by constructing a directed edge from a transaction to both of the two prior transactions that it verifies can also be visualised on this graph. One such visualization is provided in Figure 31. The spatial representation of the transactions has some useful properties. In particular, time is a reverse topological ordering of the graph by construction, so the x axis of the graph provides a valid vertex visitation order. Also, transactions are more likely to be connected if they have similar Trust Scores (y axes), especially for small values of ρ. Figure 31 illustrates that as ρ is decreases, there is less connectivity between transactions at different heights on the graph.

Figure 31: The Cluster in the space of Time by Trust Score. Empirically we find that the cluster becomes almost disconnected when $\rho \ll \lambda$.



10.2 Performance Analysis

DAG-based transaction systems have been analysed under the assumptions of a uniform random attachment algorithm with no Trust Score (e.g. [13]). These analyses found that the number of source transactions in the DAG should approximately approach the constant value $2\lambda\Delta t$ once the system has stabilised. A similar result can be obtained in our case for the number of sources.

Consider the case where there are S source transactions at some point in time, and let s be one of the current sources that will be selected by the new transaction that arrives. Before that new transaction publishes the selection to the network, it must perform validation and proof-of-work. This takes Δt time. During this time, s is still visible as a source transaction to all new transactions that arrive in the network. On average, there will be $\lambda\Delta t$ such transactions. Each of the new transactions will only be able to attach to s if it has a close enough Trust Score. Assuming that the Trust Score distribution for new transactions is identical to that of sources in the Cluster, the new transaction will have probability ρ of being allowed to attach to s, and there will be on average ρS transactions available for it to connect to. Since each new transaction attaches to two sources, we can compute the probability that s is selected

by a transaction as:

$$\rho\left(\frac{1}{\rho S}+\left(1-\frac{1}{\rho S}\right)\frac{1}{\rho S-1}\right)=\frac{2}{S}$$

Recalling that s has already been selected by the next new transaction that arrives (and therefore already has 1 transaction attached to it), this means that the expected number of transactions attaching to s is $N_A = 1 + 2\lambda\Delta t S^{-1}$. This is the average number of DAG edges that it takes to validate one source transaction. We also know that each new transaction adds two edges to the DAG before becoming a source transaction.

The quantity N_A therefore determines if S will grow or shrink over time. If $N_A > 2$, then each new transaction is removing less than one source on average and S will increase. Conversely if $N_A < 2$, then each new transaction is removing more than one source on average and S will decrease. However, since the number of sources S is in the denominator of this expression, $N_A = 2$ is an *attractor*: if $S > 2\lambda\Delta t$, then $1+2\lambda\Delta t S^{-1} < 2$ and the number of sources will decrease, and if $S < 2\lambda\Delta t$, then $1+2\lambda\Delta t S^{-1} > 2$ and the number of sources will increase. Therefore over time, S must approach the fixed point $2\lambda\Delta t$. In particular, for a sufficiently large amount of time, S can be assumed to be approximately constant.

We now consider the number of validated, unconfirmed transactions at time t, which we will denote by V. We will use the fact that after a sufficient amount of time, the expected in-degree of all transactions is 2 since every transaction starts out as a source transaction and will have an expected in-degree of 2 in the steady state.

At a fixed time, let $\mathscr{S}$ be the set of source transactions; $\mathscr{V}$ be the set of validated (unconfirmed) transactions; $\mathscr{V}_i$ the set of vertices in $\mathscr{V}$ with the longest reverse oriented path (in a number of transactions) to $\mathscr{S}$ of precisely i. We observe that $\mathscr{V} = \bigcup_{i=1}^{Ld} \mathscr{V}_i$ since every path length that is greater than Ld has a cumulative trust score greater than Ld, and so any vertex at the beginning of such a path is confirmed and not in $\mathscr{V}$.

Now the parents of any vertex from $\mathscr{V}_1$ are in $\mathscr{S}$. Further, each vertex in $\mathscr{S}$ contributes two out-edges to the DAG, and by the argument above each vertex in $\mathscr{V}_1$ consumes on average 2 vertices from the DAG, so we find that $\mathbf{E}[|\mathscr{V}_1|] \leq \mathbf{E}[|\mathscr{S}|] = 2\lambda\Delta t$. Similarly, the parents of vertices in $\mathscr{V}_2$ are in $\mathscr{V}_1 \cup \mathscr{S}$, and so $\mathbf{E}[|\mathscr{V}_2|] \leq 4\lambda\Delta t$. Proceeding inductively, $\mathbf{E}[|\mathscr{V}_i|] \leq 2i\lambda\Delta t$. Adding all of these together, we find that:

$$V = \mathbf{E}[|\mathscr{V}|] \leq (Ld)(Ld+1)\lambda\Delta t$$

This is not a strict bound, but a constant in time. We have therefore established that V is bounded above by a constant after a sufficient amount of time has passed.

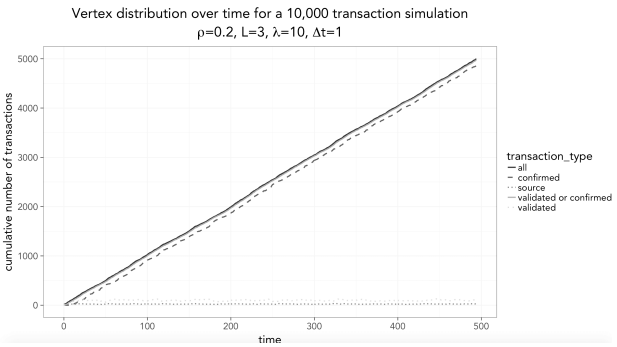
We have established that the number of source transactions S and the number of validated, unconfirmed transactions V are both approximately constant after enough time has passed. We therefore turn our attention to the number of confirmed transactions, C. Since every transaction already attached to the Cluster is either a source transaction, a validated (unconfirmed) transaction or a confirmed transaction, the total number of attached transactions in the Cluster is $S+V+C$. Taking the expected ($\mathbf{E}$) rate of change we get:

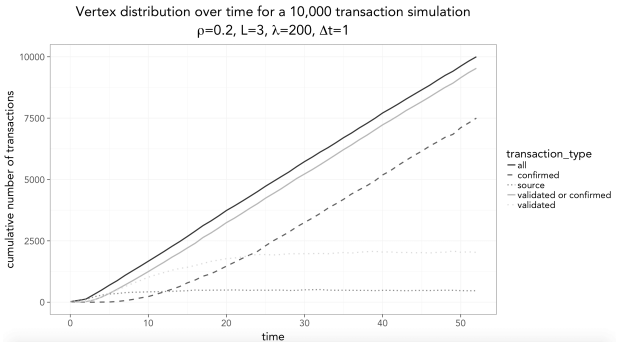
$$d[\mathbf{E}[S+V+C]] = d[N_{transactions}] = \lambda$$

It follows that because S and V are constant in expectation after sufficient time has passed, then we must have $d[\mathbf{E}[C]] = \lambda$. In other words, the rate of confirmation of new transactions matches the arrival rate of transactions.

Empirical investigations confirm that this behaviour is indeed the case, as the confirmation rate matches the arrival rate after some initial time (see Figure 32).

Figure 32: These figures show that after the initial phase, the system becomes stable and the rate at which transactions are confirmed equals the rate at which new transactions arrive. In particular, the number of sources and number of new (not yet validated) transactions becomes constant.





This provides conclusive evidence that the Cluster is scalable. The only theoretical limitation to the throughput of COTI is the number of transactions arriving per second. Because COTI was designed to be attractive to a large pool of merchants and consumers, we are confident that the number of transactions arriving per second will eventually exceed 10,000.

10.3 Simulations

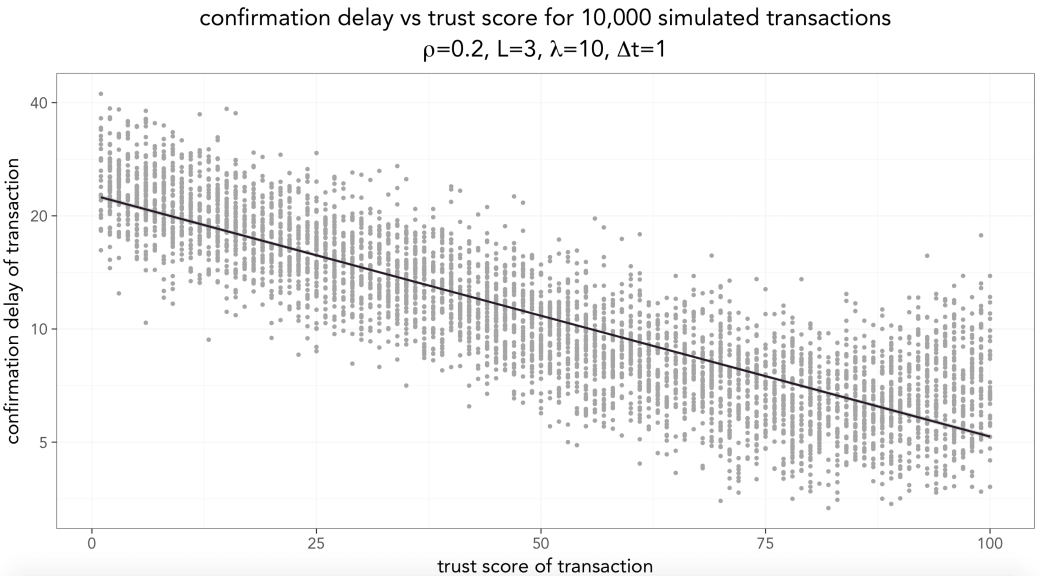
The previous section established the throughput characteristics of the Cluster under simplified conditions. To investigate other characteristics of the Cluster and to analyze more complex scenarios, we have utilised simulations.

A simulator of the Cluster was built to facilitate further analysis, and will be made available on GitHub. While the simulator is capable of constructing complex scenarios, we have in this section restricted our attention to the limited set of assumptions outlined in the mathematical framework above.

The simulation was first used to verify that the transformation $t \to ct; \Delta t \to c\Delta t; \lambda \to \lambda/c$ is a symmetry of the Cluster. This essentially means that we can rescale our time unit from seconds to "multiples of Δt" without loss of generality. Some graphs from this initial investigation are shown in Appendix A. In our remaining analyses, we set $\Delta t = 1$.

Whilst the previous section addresses confirmation throughput, it doesn't take the amount of time a transaction has waited into account. We expect that confirmation times will decrease in line with increasing Trust Score and has been confirmed by simulation results. (Figure 33).

Figure 33: The confirmation delay (log-scale) decreases markedly with increasing Trust Scores, with an order of magnitude difference between Trust Score 1 and 100. The drop-off is linear almost everywhere with flattening at the end.



11 Possible Attacks

11.1 Double-Spend Prevention

A payment solution cannot be open to the possibility of double-spending attacks. To mitigate this risk, COTI deploys dedicated Double Spend Prevention (DSP) Nodes. These nodes carry out additional transaction monitoring without affecting the network. Please refer to section 8 ("Double Spend Prevention and DSP Consensus") for details.

11.2 Penny-spend Attack/Transaction Flood

If an account is trusted, its transaction confirmation speed, PoW requirements and fees will be low. An attacker with a highly trusted account can therefore send many valid transactions with small amounts to waste the storage resources of Nodes. The PoW required to validate each transaction, however, limits the number of transactions that an attacker can send due to the high computational resources required to launch such an attack. In the rare event where an attacker is capable of making so many small transactions, the account's Trust Score will decrease, causing an increase in the PoW required to create new transactions.

Another possible way to launch a similar attack is to flood the network with invalid transactions using many user accounts. In COTI, such attempts will be met with little success because all transactions are verified by Full Nodes.

11.3 Sybil Attack

An attacker can draw up multiple resources in the form of computers, virtual machines, and IP addresses in order to create numerous accounts with different usernames and email addresses. These accounts, known as Sybil identities, can be used to subvert the use of trust in the network [10]. Since accounts with low trust can be created by newcomers, an attacker with many accounts could try to create a subcluster that begins with a double-spend from one account at the beginning of their subcluster and then proceed to validate their own transactions with other accounts by ignoring COTI's Source Selection Algorithm.

According to COTI network structure, attempting such an attack would require the attacker to run at least one malicious Full Node to maintain its subcluster, otherwise it will not be possible to ignore COTI's Source Selection Algorithm.

If a long enough chain is created, the attacker could claim to have confirmed transactions once enough trust has been accumulated. In this situation, DSP Nodes will check if such a situation has arisen and will prevent the possibility of double-spending. As a result, an attacker will pay the network fee, but will never have DSP consensus.

11.4 Man-in-the-middle Attack

Since packets can be inserted into communication channels by an attacker, the attacker may try to impersonate one of the special Nodes, such as the DSP Node. The possibility of such an attack is problematic because when a user first joins COTI they will not know if they are using the public key of an attacker or a real COTI Node. To solve this problem the COTI client will have the public key of the COTI servers hard-coded in it. The COTI servers will therefore serve a similar function to a certificate authority in an SSL/TLS handshake. Once a secure connection with the COTI servers is established, the servers will then be used to get the authentic public keys of special Nodes.

11.5 Malicious Node Attack

The fact that participants can purchase the Nodes that perform verification, namely the DSP Nodes, means that an attacker can attempt to buy favor in the COTI network. In particular, it seems that at first glance that all an attacker needs to verify their own transactions are, at minimum, two Nodes consisting of a Full Node and DSP Node.

Since verification requires consensus among DSP Nodes, however, and purchasing DSP Nodes are expensive, we assume that purchasing the majority of DSP Nodes would not result in a profitable attack. Furthermore, only participants with a high Trust Score can create a DSP Node and as soon as the Node is found to be acting maliciously by other DSP Nodes, it will lose all trust, be blocked and the owner's deposit will be seized.

11.6 Denial of Service (DDOS) Attacks

COTI is a decentralised network implementing distributed ledger technology. By design, this network has no single point of failure liable to DDOS attack. There are too many Full Nodes in the network for any imaginable DDOS attack. Less numerous DSP Nodes have load balancing and cloud-level DDOS protection systems in addition to Trust Score Nodes.

The only imaginable DDOS attack point is through the KYC servers, but these servers are important insofar as they are an entry point for new users. If new registrants are kicked off due to a DDOS attack for several hours or even days, the network will continue to work as usual.

11.7 Distribution of Software Patches

Flaws have been found in many cryptocurrency implementations [18]. Patches should therefore be distributed securely and quickly to prevent Nodes from being compromised. In the event of such a breach, any flaws found in the COTI client are unlikely to result in significant losses since balances and network transaction history are verified by DSP Nodes and History Nodes respectively.

12 Future development

In order to provide a completely decentralised ecosystem for online payments, COTI is exploring various alternatives to enable decentralised governance. This governance structure will be responsible for implementing decisions that impact the base protocol, the future use of COTI tokens, investments and more. This governing body will not only vote on such matters, but will also be responsible for executing the changes they vote for. Futarchy is one such type of governance currently being explored [8].

One example of decentralised governance is characterised by the process that might be adhered to when a protocol update is ready. Once the protocol update is ready, a team of experts in the field will create a metric to determine possible outcomes. After this is established, COTI token holders will be able to vote for the decision they think is best for the network. This means decisions will be based on the wisdom of the crowd. The mechanism for choosing the team of experts will be determined in future iterations of the network.

To streamline COTI's future development, COTI's transaction bundles will have free space set aside on which future data layers can be stored. These layers may be used by other companies that wish to deploy smart contracts and information over the Cluster, or for the purpose of transferring other currencies and data types across the network.

References

[1] Peter Arntz. Blockchain technology: not just for cryptocurrency. https://blog.malwarebytes.com/security-world/technology/2017/12/blockchain-technology-not-just-for-cryptocurrency/, 2017.

[2] blockchain.info. Confirmed Transactions Per Day. https://blockchain.info/charts/n-transactions?timespan=all.

[3] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*. Elsevier Science Ltd, 1976.

[4] Anton Churyumov. Byteball: A decentralized system for storage and transfer of value. 2016.

[5] Thomas H Cormen. *Introduction to algorithms*. MIT press, 2009.

[6] Kyle Croman, Christian Decker, and Ittay Eyal Eyal. On scaling decentralized blockchains. 2016. 20^{th} international conference on Financial Cryptography and Data Security 2016.

[7] Brian S Everitt, Sabine Landau, Morven Leese, and Daniel Stahl. *Cluster Analysis*. Wiley Online Library, 2011.

[8] Robin Hanson. Shall we vote on values, but bet on beliefs? http://mason.gmu.edu/~rhanson/futarchy2007.pdf/, 2007.

[9] Investopedia. 51% attack. https://www.investopedia.com/terms/1/51-attack.asp, 2017.

[10] Brian Neil Levine, Clay Shields, and N. Boris Margolin. A survey of solutions to the sybil attack. 2006.

[11] A.P. Moller Maersk. Maersk and IBM to form joint venture applying blockchain to improve global trade and digitise supply chains. https://www.maersk.com/press/press-release-archive/maersk-and-ibm-to-form-joint-venture, 2018.

[12] Jelena Mirkovic and Peter L. Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *Computer Communication Review*, 34, 2004.

[13] Serguei Popov. The tangle. 2017.

[14] MIT Technology Review. Blockchain primer. https://www.technologyreview.com/collection/blockchain-primer/, 2018.

[15] Yary Ribero and Daniel Raissar. Dagcoin whitepaper. 2015.

[16] Sheldon M Ross. *Introduction to probability models*. Academic press, 2014.

[17] Mustafa Suleyman and Ben Laurie. Trust, confidence and verifiable data audit. https://deepmind.com/blog/trust-confidence-verifiable-data-audit/, 2017.

[18] Bitcoin Wiki. Common vulnerabilities and exposures. https://en.bitcoin.it/wiki/Common_Vulnerabilities_and_Exposures.

Appendix A Simulation Results

A Cluster simulator has been built and will be made available on GitHub. The primary purpose of the simulator is to analyse parameter and algorithm choices and to collect empirical data within a sandbox so as to optimise real world Cluster performance. The simulator can be used to analyse the impact of internal and external parameters on the performance of the Cluster. It is accompanied by a collection of data extraction and visualization tools that enable rapid scenario analysis. The core simulator is written in C++ with analysis components in R. It is able to simulate about 1000 transactions per second on a laptop computer and has been tested in simulations with up to 5,000,000 transactions.

Some selected simulation results are displayed here, while those relevant to the discussion on performance characteristics have already appeared in this document in Section 10. Below is a summary of the relevant parameters used within most of the simulations presented here:

1. Δt is the fixed amount of time for a node to run the Source Selection Algorithm and to perform proof-of-work.
2. λ is the rate of new arrivals, which are assumed to follow a Poisson process.
3. K is the number of new arrivals that arrive in the time taken to run the Source Selection Algorithm and perform proof-of-work.
4. L is the multiplier which determines the cumulative trust threshold that a Trust Chain must surpass in order to be confirmed. The cumulative trust of a Trust Chain should exceed $100L$ for the transaction to be confirmed.
5. ρ is the width of the Trust Score threshold that a transaction can confirm. This is expressed as a fraction of the total number of new transactions.

Figure 34: Waiting time as a function of Trust Score. Note that confirmed transactions are clustered around the region of high trust and low confirmation time.

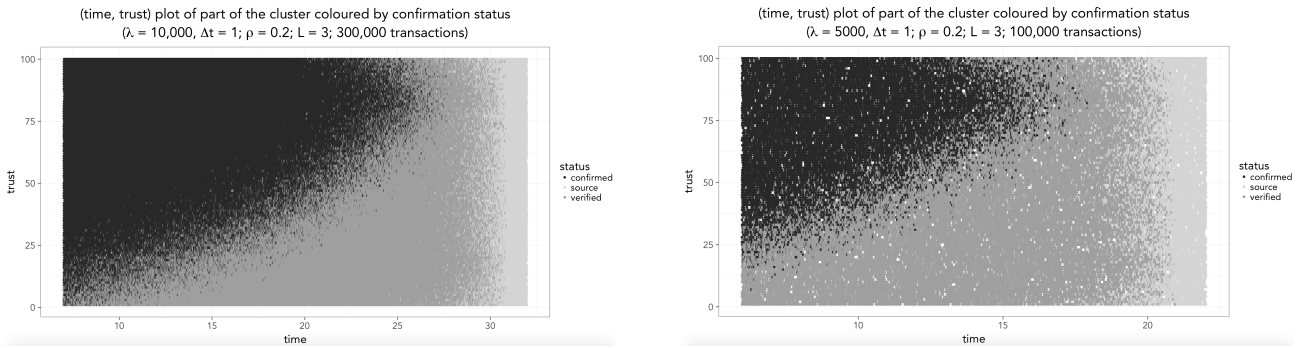


Figure 35: These figures suggest an inflection point slightly before $\lambda = 10$. Note that confirmation delay closely resembles validation delay, with the exception of trust drift.

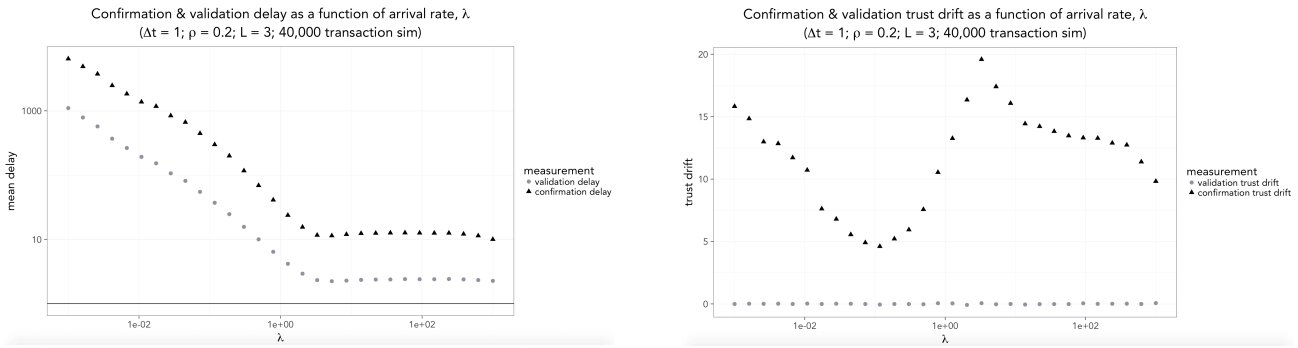


Figure 36: Note the inflection point slightly before $\lambda = 10$. This appears to be universal and is likely related to the value of λ necessary to stop the DAG from becoming disconnected as is visible in Figure 31

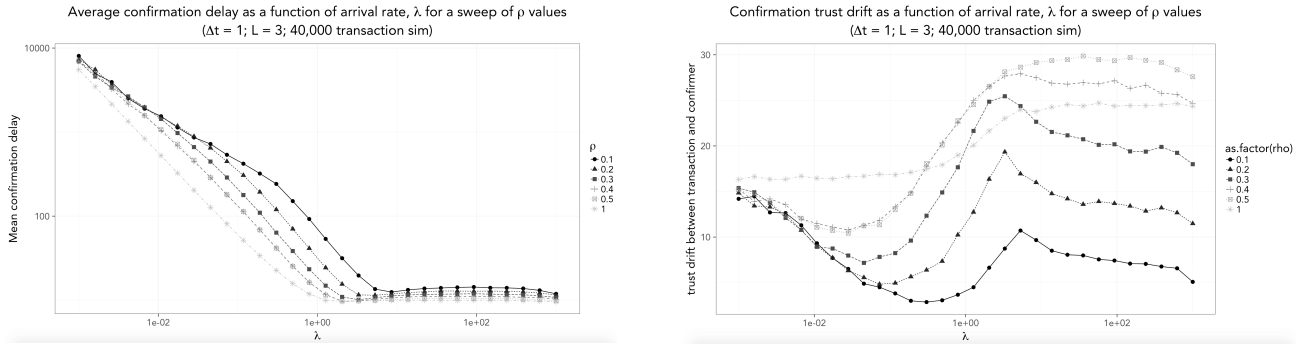


Figure 37: The figure on the left shows the confirmation time is barely affected by the size of the threshold ρ. The figure on the right shows that the mean delay in validation and confirmation times are directly proportional to the time taken to run the Source Selection Algorithm and to do the proof-of-work (Δt).

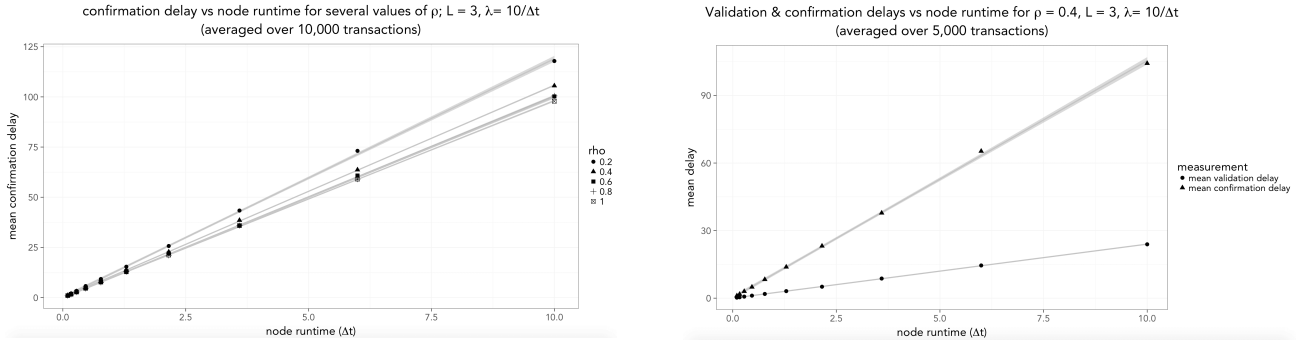


Figure 38: These figures show that the mean confirmation delay is significantly controlled by the parameter K. This means that as long as the rate of new transactions arriving is high enough, a complex proof-of-work is not detrimental to transaction throughput.

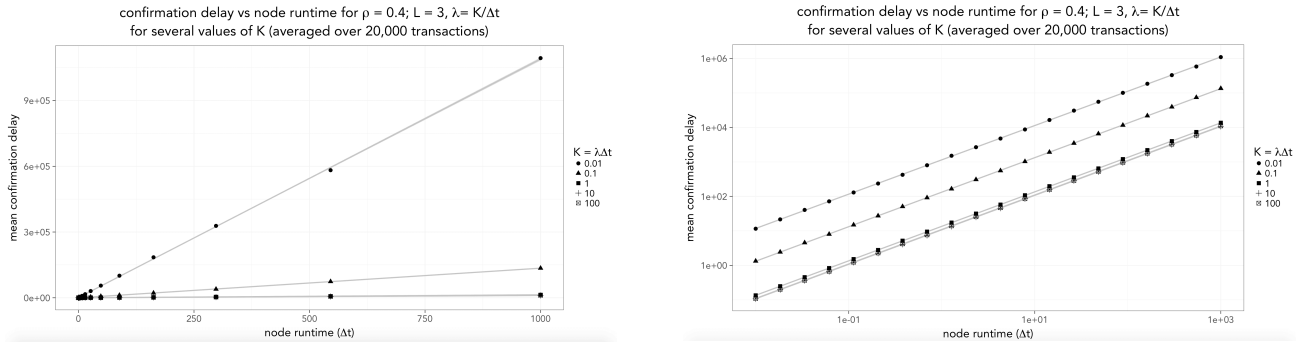


Figure 39: As can be see from this figure, the size of the threshold ρ has a definite affect on the trust drift between transaction and confirmation along the Trust Chain.

Appendix B COTI's Arbitration System

B.1 Introduction

The world is experiencing an accelerated pace of globalisation and digitalisation. An exponentially growing number of transactions are being conducted online between people across jurisdictional boundaries. If the blockchain promise comes to fruition in a not so distant future, most goods, labour and capital will be allocated through decentralised global platforms. Disputes will certainly arise, as users of a decentralised eBay will claim that sellers failed to send the goods, guests in a decentralised Airhub will argue that the rented house was not as-advertised, and backers in crowd-funding campaigns will demand a refund as teams failed to deliver on their promises.

Existing dispute resolution technologies are too slow, too expensive and too unreliable for a decentralised global economy operating in real time. A fast, inexpensive, transparent, reliable and decentralised dispute resolution mechanism that renders ultimate judgements about the enforceability of disputes is a key institution for the blockchain era.

B.2 Schelling Point

Game theorist Thomas Schelling developed the concept of Schelling Points (also known as Focal Points) as a solution that people could use to coordinate their behaviour in the absence of communication. Ethereum founder Vitalik Buterin proposed the creation of the Schelling Coin as a token that would align telling the truth with economic incentives. If we wanted to know if it rained in Paris this morning, we could ask every owner of a Schelling Coin: "has it rained in Paris this morning? Yes or no?" Each coin holder would vote by secret ballot and the results would be revealed after all parties have voted.

Parties who voted as the majority would be rewarded with 10% of their coins. Parties who voted differently from the majority would lose 10% of their coins. Thomas Schelling described Focal Points as each person's "expectation of what the other expects him to expect to be expected to do". The Schelling Coin uses this principle to provide incentives to a number of agents who do not know or trust each other to tell the truth.

We expect agents to vote the true answer because they expect others to vote the true answer, and in this case, the Schelling Point is honesty. Schelling Coin mechanisms have been used for decentralised oracles and prediction markets for decades. The fundamental insight is that voting coherently with others is a desirable behaviour that should be incentivised. The incentives design underlying the COTI Arbitration Layer is based on a mechanism similar to the Schelling Coin, but slightly modified in order to answer to a number of specific challenges regarding scaling, subjectivity and privacy to encourage agents to engage in trustworthy behaviour.

Table 4: Payoff table for a basic Schelling Game.

Majority Vote \ Your Vote	**Yes**	**No**
Yes	+0.1	-0.1
No	-0.1	+0.1

B.2.1 Arbitration and Game Theory

Arbitration is a process in which an impartial third party seeks to help two or more disputants or negotiating parties to reach an agreement. This is usually done by hosting a meeting with numerous discussions to explore the real underlying issues between them, to build proper understanding and to encourage the exchange of information between them. The parties can then identify and assess their options and alternative courses of action before reaching a mutually acceptable agreement, or ending negotiations. Experienced negotiators will help parties to measure options and consider proposals for reaching an agreement against objective standards, often called best or worst alternatives to negotiated agreement (BATNAs or WATNAs).

Arbitration tends to be a speedy and relatively cost effective process for those involved, particularly in commercial matters. The process is usually confidential and arbitrators may not, without permission, disclose to one party information given to them by another party. The entire process is usually designed to be private and without prejudice so that, in principle, no one may use information or waive any rights or remedies until the parties agree to do so, usually in writing.

Game theory is an area of study that deals with interactions where the choices of one agent influence the outcome of another, and vice versa, according to some fixed rules. Game theory attempts to predict, understand and explain activities as diverse as pricing strategies of firms, lobbying of political parties, and a couple's choice of evening entertainment. Applied initially to economics, but now prevalent throughout the social sciences and in evolutionary biology, work in this field is characterised by its abstract and mathematical approaches and its emphasis on finding common structures among diverse social phenomena.

We believe that the focus of much analysis of the value of arbitration is on the ability of the arbitrator, and the arbitration process, to enable parties to come closer to what a calm, reflective and rational negotiator would achieve. In other words, the process helps parties to separate people from the problem and to overcome cognitive biases, such as reactive devaluation and attribution error, which so often plague traditional negotiations where parties are motivated to protect their positions and reduce the risk of making unnecessary concessions. Undoubtedly, arbitration adds a huge amount of value in this dimension by helping parties communicate more effectively, avoid protracted negotiations or costly court procedures, and maintain (or even enhance) personal and commercial relationships in the process.

However, game theory suggests that arbitration could add value compared to pure negotiation, even when the parties are supremely rational, wholly self-interested agents, subject to none of the cognitive biases and other such psychological (or apparently irrational) impediments to negotiation that permeate everyday life. We believe that this aspect of arbitration's value is relatively underplayed and under-discussed, at least in some forums. We view it is as a fruitful area for arbitrators and those interested in arbitration to explore. The experiences of arbitrators could be brought together with more theoretical approaches to give richer understanding of this side of arbitration.

A range of literature in the rational-actor paradigm of traditional game theory has asked the question: 'How is it that arbitration can add value?' To many, this would appear to be an odd question to ask. But to the game theorist, it is natural. In this context of supreme rationality, why couldn't any offer that a arbitrator communicates on behalf of a party be equally well communicated by the party directly?

And, if there is no role for the arbitrator to help the super-rational parties explore all the options, weigh costs and benefits, and avoid cognitive traps, what then can the arbitrator add? Why not dispense with the arbitrator altogether?

The answer to this question has proved to be more complex than some game theorists first supposed. Actually, research predicts that a arbitrator can add value relative to a pure negotiation process between rational actors by helping parties to overcome one of the fundamental challenges in negotiation: generally, by applying the 'BATNA' yardstick, parties know what they would be willing to settle on, but they don't know what the 'BATNA' is for their negotiating counterpart. This is the origin of the incentive to disguise one's own true negotiating position and to resist making concessions as far as possible. This can lead to the parties failing to reach a settlement, even when there are potential agreements that would give both a better outcome than their 'BATNA'.

One way arbitrators can, in theory, help disputants overcome this possible barrier is by taking some information from parties, but transmitting only a portion of it to the 'other side'. For example, if a arbitrator commits to using his or her first exchanges with parties to establish whether the agreement might be possible (i.e. the fact that a 'zone of agreement' exists between the parties) and to break off negotiation if it is not, but not to tell the parties the specifics of what has been disclosed to the arbitrator. The incentive for parties to be strategic and bid up or down their offers is much reduced as they risk losing a deal by overplaying their respective hands.

A arbitrator can also help by administering a pre-approved process to which parties could not rationally adhere to if negotiating on their own. For example, an arrangement could be made to place a time limit on the arbitration process. A arbitrator who stands to neither gain nor lose by implementing this arrangement would be committed to doing so (and the parties would know this), even when one or both negotiating parties might be prepared to modify the time limit in order to drive a harder bargain.

Professional arbitrators see dynamics of this sort playing out regularly in reality. In a sense, the parties perform for the arbitrator and act more reasonably by virtue of his/her presence.

We think that game theory can help us better analyse aspects of the arbitrator's role, hitherto perhaps understood tacitly and pursued on instinct and experience. As theory and practical experience accumulate, there is surely much to gain from bringing together the findings of the game theoretic literature and the insights of arbitration practitioners.

B.2.2 COTI's Arbitration Service

The COTI Arbitration Service introduces a decision protocol for a multi-purpose court system able to solve every dispute type. It is COTI's autonomous system that works as a decentralised third party to arbitrate disputes, from very simple to highly complex disagreements. Every step of the arbitration process (securing evidence, selecting jurors, etc.) is fully automated, with the exception of juror decisions, once a dispute reaches disagreement between the parties.

COTI does not rely on the honesty of a few individuals, but on game-theoretical economic incentives. It is based on a fundamental insight from legal epistemology: a court is an epistemic engine, a tool for ferreting out the truth about events from a confusing array of clues. An agent (jury) follows a procedure where an input (evidence) is used to produce an output (decision). COTI leverages the technologies of crowd-sourcing, blockchain and game theory to develop an arbitrary system that produces true decisions in a secure, efficient and inexpensive way.

The COTI payment system is designed to provide users with a new level of quality for this service. COTI is a complex and comprehensive solution, incorporating many important features on the protocol level. One of the most important services provided by COTI is **arbitration**.

The COTI Arbitration Service provides users with a quick, reliable and inexpensive way to resolve disputes. This highly required feature is not possible with other cryptocurrencies, as COTI provides a ready-to-use service any customer can appeal to.

COTI's Arbitration Service is a decentralised application built on top of the COTI Trustchain that works as a decentralised third party to arbitrate disputes between buyers and sellers. It relies on game theoretic incentives in order for jurors to correctly rule cases. The result is a dispute resolution system that renders ultimate judgements in a fast, inexpensive, reliable and decentralised way.

In the COTI Arbitration Service disputes are resolved by a arbitrator jury randomly picked from a large pool of highly trusted network participants. The process of forming the arbitrator jury, decision and settlement is decentralised and cannot be biased by any party.

This document describes the complete set of rules, principles and architecture of the COTI Arbitration Service.

B.3 Principles

The COTI Arbitration Service is based on the following principles:

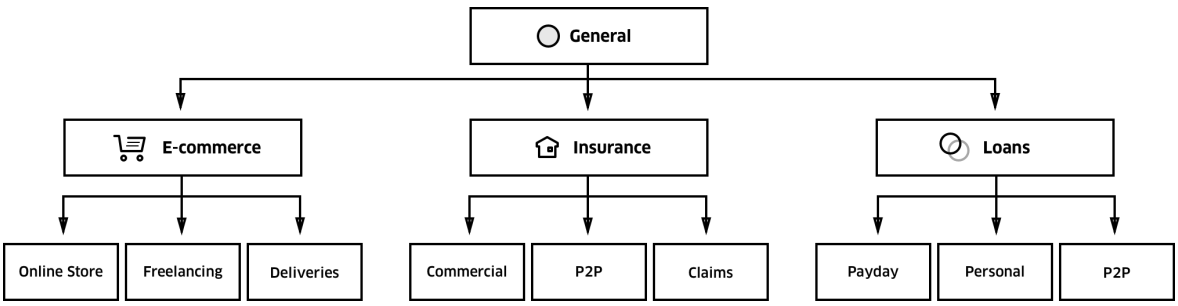
- Fairness
- Justice
- Voluntariness
- Equality
- Decentralisation
- Predictability
- Timely resolution

B.4 Project description

B.4.1 Arbitration Process

Arbitration in COTI is characterised by an embedded court system within the network. The idea is that users can choose the type of court specialised in the topic of the dispute they have lodged. A software development dispute will choose a software development court or jury, while an insurance dispute will select an insurance court or jury, for example. Figure 40 illustrates an example of the court arborescence from which users can choose.

Figure 40: Court arborescence which users can choose from.



B.4.2 Privacy

Solving disputes may require parties to disclose privileged information with jurors. In order to prevent outside observers from accessing this information, the natural language (English or other) and the labels of the jurors' voting options are not stored on the ledger. When the dispute is created, the creator submits hash(dispute text, option list, salt) 2 (where dispute text is the plain English text,

option list the labels of the options which can be voted by jurors. Salt is a random number to avoid the use of rainbow tables).

The dispute creator sends {dispute text, option list, salt} to each party using asymmetric encryption. In this way, parties can verify that the submitted hash corresponds to what was sent to them. In the case of a dispute, each party can reveal {dispute text, option list, salt} to jurors who can verify that they correspond to the hash submitted. They can do so by using asymmetric encryption such that only the jurors receive the text of the contract and of the options. All these steps are handled by the application or wallet that the users run while using the COTI Arbitration Process.

B.4.3 Drawing jurors

Users have an economic interest in serving as jurors in COTI, as they collect an arbitration fee for their work. The probability of being drawn as a juror is randomly set, which means that for a specific dispute, the amount of tokens a juror stakes is not related to his/her probability of being chosen as a juror.

The higher the amount of tokens a juror stakes, the higher the gain from voting with the majority. Jurors that do not stake COTIs will not have the chance of being drawn as a jury, which will prevent inactive jurors from being selected.

COTI stakes serve two key functions in the Arbitration Service design. First, they protect the system against sybil attacks. As such, a malicious party will not be able to create a high number of addresses to be drawn a high number of times in each dispute. This is because the arbitrator onboarding process requires completing the KYC process, as well as having many accounts with high Trust Scores.

Moreover, the proof-of-stake concept in the COTI Arbitration System works in a way that ensures any juror in the network is incentivised for his/her work. The percent of fees received is also in direct relation to the stake and past record voting.

B.4.4 Votes

After assessing the evidence, jurors commit their votes. They submit a hash(vote). When the vote is over, they reveal {vote,salt}, and COTI verifies that it matches the commitment. Jurors failing to reveal their vote are penalised. After jurors have made a commitment, their vote cannot be changed. But it is still not visible to other jurors or the disputants. This prevents a juror's vote from influencing the vote of other jurors. Jurors can still declare that they voted in a certain way, but they cannot provide other jurors a reason to think that what they say is true. This is an important feature of the Schelling Point because if jurors knew the votes of others jurors, they could vote like them instead of voting for the Schelling Point.

Jurors are also required to provide a justification for their vote. Jurors that fail to reveal their vote are penalised. Finally, votes are aggregated and the resolution of the dispute is executed. The option with the highest amount of votes is considered the winning one.

B.4.5 Arbitration fees

Creating dispute cases requires arbitration fees In order to compensate jurors for their work and to avoid having attackers spam the system. Each juror will be paid a fee determined by the dispute amount and his/her stake. The arbitration fee is taken from the rolling reserve.

Further examples include:

- In the first instance, each party will deposit an amount equal to the arbitration fee. If one party fails to do so, it will be regarded that the court ruled in favor of the party who deposited the arbitration fee (without even creating a dispute in the court). If both parties deposit the funds, the winning party will be reimbursed when the dispute has been resolved.

- In appeals, both parties have to deposit the arbitration fees. The appellant also has to deposit an extra stake proportional to the appeal fees that will be given to the party winning the dispute.

In this way if a party makes frivolous appeals to harm the opposing party, the opposing party will get a compensation for the time loss. If the appeals are ruled to be legitimate, the stake will be returned to the appellant.

B.4.6 Appeals

If, after the jury has reached a decision, a party is not satisfied because it thinks the result was unfair, it can appeal and have the dispute ruled again. Each new appeal instance will have twice the previous number of jurors plus one. Due to the increased number of jurors, appeal fees must be paid.

If a verdict is appealed, jurors of the appealed level are not paid, but are still affected by the dispute due to token redistribution. This incentivises jurors to give explanations of their rulings. When proper explanations are given, parties are less likely to appeal as they have a lesser likelihood to be convinced that a decision is fair.

Due to arbitration fees being paid to each juror and appeals increasing the number of jurors exponentially, arbitration fees rise in line with the number of appeals. This means that, in most cases, parties won't appeal, or will only appeal a moderate amount of times. However, the possibility of appealing a high number of times is important to prevent an attacker from bribing jurors.

B.4.7 Incentive system

Jurors rule disputes in order to collect arbitration fees. They are incentivised to rule honestly because after a dispute is over, jurors whose vote is not coherent with the group will lose arbitration fees that will be given to coherent jurors. After the COTI Arbitration System has reached a decision on the dispute, tokens are unfrozen and redistributed among jurors. The redistribution mechanism is inspired by the Schelling Coin, where jurors gain or lose arbitration fees depending on whether their vote was consistent with the other jurors.

Small disputes are defined as training disputes for new arbitrators for which the network will favour assignment, while still making sure that at least one veteran arbitrator is added to the poll.

We will assume a jury member voted coherently if he/she voted for the option chosen by the majority. The amount of tokens lost from the arbitration fee per incoherent juror is: $\alpha \times min\ activate \times weight$. The α parameter determines the number of tokens to be redistributed after a ruling. It is an endogenous variable that will be defined by the governance mechanism as a consequence of the internal dynamics of the voting environment. The *min activate* parameter is the minimum amount of tokens that can be activated in the dispute. The arbitration fees are divided between the coherent and incoherent parties proportionally to their weight. Parties are considered coherent if they voted as the majority.

Figure 41: Example of token redistribution.

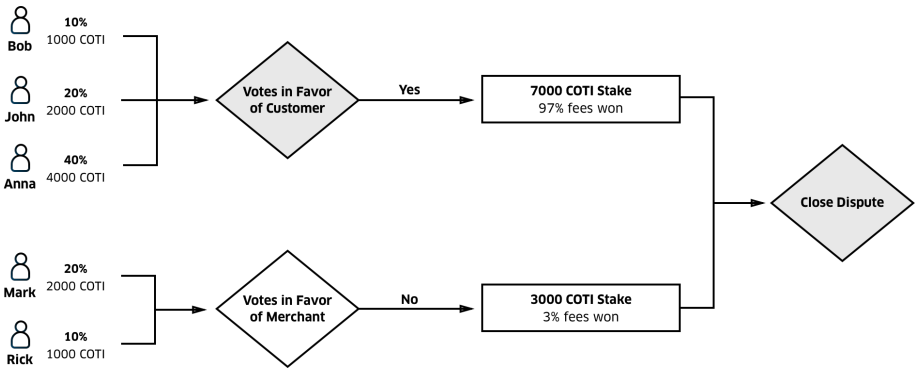


Table 5: Example of the distribution of $100 dispute fees.

Arbitrator	**Stake in COTI**	**Vote**	**Fees Won**
Bob	1000	Yes	$10.38
John	2000	Yes	$20.77
Anna	4000	Yes	$50.54
Mark	2000	No	$1
Rick	1000	No	$2

The fee percentage of each arbitrator is calculated as follows:
Given:
F – the dispute fee
S_i – The stake of participant i
$\mathbb{P}$ – Number of arbitrators voting
ω – Number of arbitrators voting with the majority
$\bar{\omega}$ – Number of arbitrators voting with the minority
κ – Ratio of fees deducted from minority (in percent)
$K = F \cdot \kappa$ – Fees distributed to the minority
$W = F - K$ – Fees distributed to the majority
$S_n = \sum_{i=0}^{\mathbb{P}} S_i$ – The total stake given for the dispute
$S_\omega = \sum_{i=0}^{\omega} S_i$ – Total stake of those voting with the majority
$S_{\bar{\omega}} = \sum_{i=0}^{\bar{\omega}} S_i$ – Total stake of those voting with the minority
$W_\omega = \frac{S_\omega}{S_n} \cdot W$ – Winning fees for the majority
$W_{\bar{\omega}} = \frac{S_{\bar{\omega}}}{S_n} \cdot K$ – Winning fees for the minority

Jurors could fail to reveal their vote. To disincentivise this behaviour, the penalty for not revealing one's vote is twice as large as the penalty for voting incoherently ($2 \cdot \alpha \cdot minactivate \cdot weight$). This incentivises jurors to always reveal their vote. In case of appeals, the tokens are redistributed at each level according to the result of the final appeal. When there is no attack, parties are incentivised to vote what they think other parties think is honest and fair. In COTI, the Schelling Point equates to honesty and fairness. One could argue that these decisions being subjective would not enable a Schelling Point to arise.

The informal experiments run by Thomas Schelling showed that in most situations a Schelling Point plebiscite by all parties does not exist. But Schelling found that some options were more likely to be chosen than others. Therefore, even if a particularly obvious option does not exist, some options will be perceived as more likely to be chosen by others parties and will effectively be chosen. We cannot expect jurors to be right 100% of the time – no arbitration procedure could ever achieve that. Sometimes, honest jurors will lose arbitration fees, but as long as they lose less value than what they win as arbitration fees for other incoherent parties, the system will work.

Arbitrators are incentivised to participate in the dispute resolution by the arbitration fee. This fee depends on the stake in COTI coins by the arbitrator on the particular case.

The most effective incentive for arbitrators to be fair is the pursuit of justice, which is instinctive human nature.

The main motivation problems to be dealt with are first: the "lazy" strategy , that is to not read case data, provide a random vote and earn a fee; and second: the possibility of a biased opinion based on affiliation, nationality, religion, culture, gender, etc.

To mitigate these problems, COTI will implement the following measures:

1. Arbitrators are highly trusted network participants according to the Trust Score metrics. High Trust Score value means that if the arbitrator is successful in a societal activity, so we may suppose that he/she is a responsible person.
2. Arbitrators are randomly chosen.
3. There will be a control set of questions automatically generated from the case data to check that the arbitrator completed them.
4. There will be an AI-based system to analyse arbitrator votes to detect possible biased or "lazy" arbitrators.
5. Depersonalisation of case data to the maximum extent possible.

B.4.8 Dispute overview

When it comes to filing and resolving complaints, COTI will treat both sellers and buyers fairly, while guiding both through the resolution process.

There is only one way that a buyer can initiate a complaint. That process, including time frames and who is in charge of settling the dispute, will vary depending on how the purchase was funded.

- Dispute/claim: Buyer contacts the seller directly through the COTI wallet arbitration interface to file a dispute, and the two parties work together to find a solution. If the buyer and seller cannot agree to a solution, the buyer can escalate the dispute to a claim in order to request a refund/reversal. The arbitration process steps in to determine how the situation should be resolved, and the buyer will need to provide proof of evidence to support his/her claim.

If a buyer initiates a complaint, the seller will receive an email about it and will also see it in the arbitration tab of his wallet. Additionally, the money the seller received for the transaction may be unavailable during the case investigation. If the case is settled in the seller's favour, the money will be released back to his/her COTI account. Read further for an in-depth look at the various types of complaints, how to resolve them and how they can be avoided:

Table 6: Summary of the types of objections.

Type of Resolution	**Who initiates the case?**	**Who determines the case outcome?**	**Who owns the process/ policy?**	**Is there a processing fee?**	**What types of issues?**
Dispute	Buyer	Buyer and seller	COTI Arbitration Process	Yes	Item Not Received (INR), Significantly Not As Described (SNAD)
Claim	Buyer or seller	COTI Arbitration Process	COTI Arbitration Process	Yes	Item Not Received (INR), Significantly Not As Described (SNAD), Unauthorised Transaction

Disputes: When buyers file a complaint through the COTI arbitration tab.

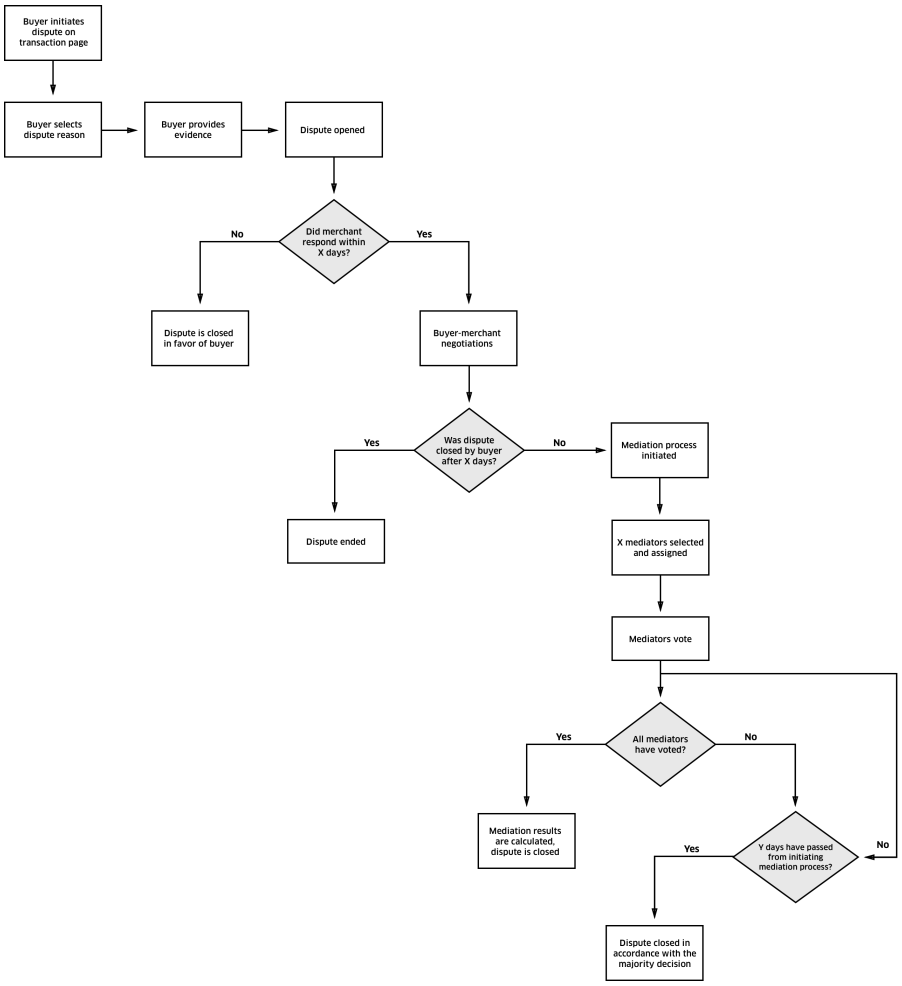
If buyers have a problem with a transaction, they can bring it to the seller's attention by opening a dispute in the arbitration tab of the COTI wallet. The dispute process is an opportunity to resolve issues before they become escalated to a claim. It's in a seller's best interest to work with the customer to resolve the dispute. This is a seller's chance to use great customer service to solve an issue and help prevent it from growing into something larger.

Why do disputes occur?

A buyer may file a dispute for three different reasons:

- Item Not Received (INR). In this case, the buyer is claiming they ordered and paid for an item, but didn't receive it.
- Significantly Not As Described (SNAD). In this type of claim, the buyer is stating that the item they received is significantly different than what they expected based on the seller's description. For instance, maybe the buyer ordered a red sweater but received a blue one instead.
- Unauthorised Transaction/Fraud. If a complaint is filed for this reason, it means the buyer's account may have been compromised or hacked and that someone made a purchase from the account without their permission. It may also mean the buyer believes that the transaction was issued without his/her consent.

Figure 42: A flow diagram of the Dispute Process



How will merchants know if a dispute has been filed against them?

If a dispute is filed against merchant, he/she will receive an email about it and a case will be created in the arbitration tab and dashboard of the merchant wallet.

How can a merchant respond to a dispute?

To respond to a dispute, a merchant should:

- Log in to his/her wallet.
- Go to the arbitration tab.
- Click View under Action next to the dispute case.
- Respond to the buyer, making sure to include any relevant information (such as package tracking information), and then select Post Message.

The message will be sent directly to the buyer. This is an opportunity to resolve the dispute without intervention (no arbitration jury selection), so it's best to be courteous and helpful.

How can a merchant help prevent a dispute? Good communication is important to help prevent disputes.

- Provide detailed, accurate descriptions of items for sale and include pictures from multiple angles.
- When a purchase is made, merchants should ship items promptly and provide tracking information.
- Send any recorded phone calls if the purchase was made via telephone.
- Signed agreements, T&C, risk disclosures etc.
- Post customer service contact information, including working hours and response time frames. A toll-free phone number can also be helpful and in some cases preferred over an email address.
- Merchant should offer a refund and post their return policy where customers can see it.
- If a customer contacts a merchant, the merchant should be professional, helpful and courteous.

What if a merchant can't reach an agreement with the buyer?

Once a dispute is opened, customers have a time window (specified per case) to work with the buyer to resolve it (arbitration won't be involved at this point). If neither the merchant nor the buyer escalates the dispute during the 14-day window, it will be closed after the arbitration process. If the buyer/seller cannot work out a resolution, either party can elevate the dispute to a claim, which we'll cover in the next section.

Each transaction made to merchants includes a rolling reserve fee that keeps 2% of total merchant turnover for a period of 6 months. Only the arbitrators have the ability to release funds to merchant accounts, and upon their decision, funds can be released back to the merchant account. Merchant rolling reserve funds are temporarily unavailable when a dispute is filed. This hold will remain in place while the merchant works with the buyer to resolve the dispute and will be released if the dispute is settled in favour of the consumer/merchant.

Claims: When a buyer complaint is escalated in the COTI Arbitration Process, or the buyer filed an unauthorised transaction.

If a buyer dispute cannot be resolved, either party can escalate it to a claim during the threshold period. At this point, the arbitration process becomes directly involved and jurors will make a decision using the information provided. A buyer can also file a claim (without first initiating a dispute) if they feel their account has been used fraudulently. During the resolution process, the Arbitration System may ask both parties for more information to be reviewed by the arbitrators.

Why do claims occur?

Claims are based on three factors:

- Item Not Received (INR). In this case, the buyer is claiming they ordered and paid for an item but didn't receive it.
- Significantly Not As Described (SNAD). In this type of claim, the buyer is stating the item they received is significantly different than what they expected based on the seller's description. For instance, the buyer ordered a red sweater but received a blue one instead.
- Unauthorised Transaction. If a complaint is filed for this reason, it means the buyer's account may have been compromised or hacked and someone made a purchase from the account without their permission.

How will a merchant know if a claim has been filed against him?

If a claim has been logged, the merchant will be notified via email. They will also see that a case has been created in the arbitration tab of the merchant wallet.

How will the claim be processed?

If a claim is filed, the seller will be asked to respond within ten days. If the seller doesn't respond, the claim will automatically close in the buyer's favour, and a full refund will be issued. If the seller doesn't respond, COTI will initiate the arbitration process in which the jurors will evaluate the information provided and determine the outcome of the claim.

Will a merchant be penalised for having claims?

Having a claim filed against a merchant doesn't necessarily mean they will be penalised. There are no automatic fees levied against merchants, and the merchant Trust Score won't automatically be affected. However, if a claim rate is too high, or other indicators are trending negatively it will affect the merchant's Trust Score.

B.5 Cases

There are different types of disputes. For each known type of dispute, the trial parameters will be defined, such as the number of arbitrators in the jury, arbitrators' training level, parties' anonymity level and fee levels.

B.5.1 Credit not processed

The customer claims that the purchased product was returned, or the transaction was canceled, although the merchant has not yet provided a refund or credit.

Required to overturn dispute: Merchant must demonstrate that the customer has been refunded through other means, or that the customer is not entitled to a refund. Merchants cannot issue a refund while a payment is being disputed. If a merchant believes that the customer was entitled to a refund that was not provided, the merchant can accept the dispute.

How to respond: Merchants should first get in touch with the customer. If the merchant understands the complaint, there will be a chance for the merchant to explain the misunderstanding or to resolve it. If the merchant is unable to solve the issue directly with the customer he/she can wait until the dispute is resolved automatically via the arbitration process.

How to prevent it: Merchants should provide a clear return policy and make it easily accessible. For customers requesting a replacement or refund, the merchant should make sure to honour the returns or refund policy promptly.

B.5.2 Duplicate

The customer claims they were charged multiple times for the same product or service.

Required to overturn dispute: Demonstrate that each payment was for a separate product or service.

How to respond: Merchant needs to determine if the customer was incorrectly charged multiple times. If they were not, the merchant needs to collect any and all information documenting that each payment was made separately, such as receipt copies. If the receipts don't include the items purchased, the merchants need to make sure to include an itemised list. Each receipt should clearly indicate that the payments are for separate purchases of items or services. If the merchant is unable to get in touch with the customer, this should be included in the case's supporting evidence.

If there were duplicate payments, merchants should accept the dispute. Merchants cannot issue a refund while a payment is being disputed. If there were two or more separate payments, merchants should get in touch with the customer. If the merchant understands the complaint, there will be a chance to explain the misunderstanding or to resolve it. If merchants are unable to solve the issue with the customer it will be directed to the arbitration process.

B.5.3 Fraud

This dispute occurs when a COTI token holder claims that they didn't authorise a payment. This can happen if the account was hacked and used to make a fraudulent purchase.

Required to overturn dispute: Get the account holder to withdraw the dispute by identifying the payment or proving to the issuer that he/she did indeed authorise it.

How to respond: First, the merchant should try to get in touch with the account holder. Sometimes people forget about payments they make. It's also possible that there was an authorised user on the account (e.g., a spouse) who made the payment.

Having the buyer withdraw the dispute is by far the best way for merchants to make sure a dispute has been resolved. If buyers agree to this, merchants should still submit evidence for the dispute. In addition, the evidence should include correspondence with the buyer stating that they will withdraw the dispute, in addition to a written statement from the buyer confirming that the dispute has indeed been withdrawn.

If merchants believe the payment was actually made fraudulently, it is better for them to accept the dispute.

B.5.4 General

This is an uncategorised dispute, so the merchant should contact the customer for additional details to find out why the payment was disputed.

B.5.5 Product or service not received

The customer claims the products or services purchased were not received.

Required to overturn dispute: Merchant needs to prove that the customer received a physical product or offline service, or made use of a digital product or online service. This must have occurred prior to the date the dispute was initiated.

How to respond: First, the merchant should get in touch with the customer. Understanding why a dispute was filed will be important for ensuring the customer receives the product and will give the merchants critical information to prevent this from happening to others.

Having the cardholder withdraw the dispute is the best way for merchants to make sure a dispute has been resolved. If an agreement between the buyer and seller cannot be reached, the dispute will be escalated to the COTI arbitration process.

B.5.6 Unacceptable product

The product or service was received but was defective, damaged, or not as described.

Required to overturn dispute: Demonstrate that the product or service was delivered as described at the time of purchase.

How to respond: First, merchants must get in touch with the customer. If they understand why they're dissatisfied, there is a chance for the merchant to explain the misunderstanding or to resolve it.

Having the buyer withdraw the dispute is the best way for merchants to ensure a dispute has been resolved.

B.5.7 Subscription cancelled

The customer claims that a merchant continued to charge them after a subscription was cancelled.

Required to overturn dispute: The merchant must prove that the subscription was still active and that the customer was aware of, and did not follow, the cancellation procedure.

How to respond: First, merchants must get in touch with the customer. If they understand what happened, there is a chance for the merchant to explain the misunderstanding or to resolve it.

Having the buyer withdraw the dispute is the best way for merchants to ensure a dispute has been resolved.

B.5.8 Unrecognised

The customer doesn't recognise the payment appearing in his/her account history.

Required to overturn the dispute: As with fraudulent disputes, assisting the customer with identifying the payment, so he/she can withdraw the dispute.

How to respond: First, the merchant must get in touch with the buyer. Sometimes people forget about payments they make. It's also possible that an authorised user on the account (e.g., a spouse) made the payment.

Having the buyer withdraw the dispute is the best way for merchants to ensure a dispute was resolved.

Table 7: Concluding matrix (for illustrative purposes only)

Use case	Dispute amount	Number of arbitrators	Window of RCF lock-up	Arbitration selection
Credit not processed	\$10-\$100	3	30 days	After 15 days
	\$100 - \$1000	5		
	> \$1000	11		
Duplicate	\$10-\$100	3	45 days	After 25 days
	\$100 - \$1000	5		
	> \$1000	11		
Fraudulent	\$10-\$100	3	45 days	After 25 days
	\$100 - \$1000	5		
	> \$1000	11		
General	\$10-\$100	3	15 days	After 7 days
	\$100 - \$1000	5		
	> \$1000	11		

Table 7 – continued from previous page

Use case	Dispute amount	Number of arbitrators	Window of RCF lock-up	Arbitration selection
Product not received	\$10-\$100	3	60 days	After 30 days
	\$100 - \$1000	5		
	> \$1000	11		
Subscription cancelled	\$10-\$100	3	30 days	After 15 days
	\$100 - \$1000	5		
	> \$1000	11		
Unrecognised	\$10-\$100	3	30 days	After 15 days
	\$100 - \$1000	5		
	> \$1000	11		

B.5.9 How evidence submission works

Each dispute has multiple parties involved in the process. Although the COTI Arbitration Layer is not involved in deciding the outcome of the dispute, it plays a role by conveying evidence to the jurors.

What to submit

The evidence submitted should be relevant to the cause of the dispute. Web logs, email communications, shipment tracking numbers, delivery confirmations, proof of prior refunds or replacement shipments can all be helpful. For example, a response to a dispute with the reason 'product not received' should have evidence that includes shipping information and any screenshots of package tracking.

When issuing evidence for disputes, requests to call or email for more information, or links to click for further information should not be included (e.g., file downloads or links to tracking information), as these will not be logged by the arbitrators who are responsible for evaluating the dispute. COTI arbitrators will not call merchants or follow external links, so it's important to submit all available evidence through the COTI Arbitration System.

Keep your evidence relevant

A long introduction about the product or company, a complaint about the customer, or the unfairness of the dispute will not make the responses more compelling. Instead, it is advisable to provide only the facts concerning the original purchase using a neutral and professional tone.

For example, John Smith purchased X from our company on [date] using his COTI wallet. The customer agreed to our terms of service and authorised the transaction. We shipped the product on [date] to the address provided by the customer and it was delivered on [date]. Merchants should also include email correspondence or texts with the customer, but it's important to be aware that these exchanges do not verify identity. If merchant's do include them, they should ensure only the relevant information is included (e.g., when including a long email thread, it is better to redact any text that is only quoting previous emails). The evidence should be factual, professional, and concise. While providing little evidence is a problem, overwhelming customers,merchants and arbitrators with unnecessary information can have a similar impact.

Provide clear and accurate evidence

The arbitrators reviewing the responses are going to decide fairly quickly whether or not the evidence is sufficient to refute the claims. For responses with multiple pieces of evidence, participants can also include a table of contents and give each uploaded image or PDF an attachment number or letter. A

lengthy terms of service or refund policy that has the relevant information highlighted can make the case significantly clearer.

Customers and arbitrators will not follow any links provided in a response. Instead, it is advised to include a clear screenshot of terms or policies as they appear during the checkout process, or on the merchant site if they are an important part of the defence (e.g., a customer is disputing a subscription, although there is a minimum contract term that must be adhered to).

Include proof of customer authorisation

Proving the customer was aware of and authorised the transaction being disputed is vitally important in any case. Any data that shows proof of this is a standard part of a compelling response, such as:

- AVS (Address Verification System) match
- Signed receipts or contracts
- IP address that matches the customer's verified billing address

Include proof of service or delivery
In addition to fraudulent disputes, claims from customers that products or services never arrived, were defective or unsatisfactory, or not as described are also potential dispute reasons. Assuming that all is well on the merchant's side (the product was not faulty, was as described, was shipped and delivered prior to the dispute date), then merchants should provide proof of service or delivery.

For purchases of physical goods, the merchant should provide proof of shipment and delivery that includes the full delivery address, not just the city and zip code. Choosing a carrier or delivery method that requires a signature on delivery provides the best defense against product not received or fraudulent disputes where merchants have shipped to a verified billing address that has passed AVS and zip code verification.

If your customer provides a 'ship to' name that differs from their own (e.g. for the purchase of a gift purchase), customers should be prepared to provide documentation explaining why they are different. While it's common practice to purchase and ship to an address that doesn't match the KYC verified billing address, this is an additional dispute risk.

If the merchant's business provides digital goods, then supporting evidence, such as the IP address or system log proving the customer downloaded the content or used the software or service should be submitted.

Include a copy of your terms of service and refund policy
Providing proof that the customer agreed to and understood the merchant terms of service at checkout, or did not follow return/refund policies is critical. A legible screenshot of how the terms of service or other policies are presented during checkout is important supporting evidence-it is not enough to simply include a text copy of these.

B.6 Decentralised governance

Arbitrators voting on decentralised governance decisions is one of the most significant functions of the COTI Arbitration System. In decentralised governance voting, all active arbitrators have one vote each. NB: This provision is subject to future changes.

B.7 Arbitrators

The COTI Arbitration System maintains a pool of reputable network participants from which the arbitrators are randomly chosen for the arbitrators jury. Users are invited to the arbitrators pool if they have maintained high Trust Scores. The Arbitration System is a decentralised human-input service, even when a particular arbitrator is a legal entity.

It is not required for network participants to deposit any amount of COTI, or maintain any amount of COTI to be invited to the arbitrators pool. However, to participate in the arbitration process it is required that arbitrators have a sufficient amount of COTI locked per their stake and released once the dispute decision has been made.

Arbitrator recruitment and training
Individuals who wish to register as arbitrators must satisfy certain requirements before being admitted to the arbitrator platform. A committee of randomly selected arbitrators will have the ability to select arbitrators and add them to the network.

Among other requirements, arbitrators must demonstrate relevant language proficiency and undergo an online assessment to determine that they have the aptitude to perform the arbitration tasks at a high standard. COTI endeavours to make arbitration open to a broad group of people and will make available online training programs that can assist candidates in acquiring the requisite knowledge to effectively contribute to the dispute resolution process. The training program will consist of:

- Online courses
- Procedures and manuals
- Online certification exams
- Internship by participating in small disputes cases

Arbitrator onboarding process
Particularly, but not restricting by this, to be compliant financial institution, COTI will implement the following KYC/AML due diligence points:

1. Customer identity.
2. Beneficial owner identity (if applicable).
3. Purpose and intended nature of business relations.
4. Ongoing monitoring, including transaction monitoring. In the EU and UK all occasional (not regular) transactions larger than 15,000 EUR should be reviewed.
5. Source of funds may be required to explain the nature of business relations and in process of monitoring.

Computer-aided dispute resolution
In subsequent versions of the COTI Arbitration Layer we would like to add AI-based tools that will help analyse case data and provide recommendations to arbitrators regarding how to judge certain disputes. This will drastically decrease the cognitive costs of arbitrators and make the arbitration process more efficient.

AI online dispute resolution tools
COTI will develop a three step model for AI online dispute resolution. Our online dispute resolution environment will be a virtual space in which disputants will have a variety of dispute resolution tools at their disposal. Participants can select any tool they consider appropriate for the resolution of their conflict and use the tools however they desire. The proposed three-step model is based on a fixed order. The system proposed conforms to the sequencing outlined below, which in our opinion produces the most effective online dispute resolution environment:

1. First, the negotiation support tool should provide feedback on the likely outcome(s) of the dispute if the negotiation were to fail (i.e., the BATNA).
2. Second, the tool should attempt to resolve any existing conflicts using argumentation or dialogue techniques.

3. Third, for those issues not resolved in step two, the tool should employ decision analysis techniques and compensation/trade-off strategies in order to facilitate dispute resolution.

By narrowing the issues, time and money can be saved in the process. Further, the disputants may feel it is no longer worthwhile to achieve their initially desired goals.

Decision support systems

Decision support systems supplement human knowledge management skills with computer-based means for managing knowledge. They accept, store, use, receive and present knowledge pertinent to the decisions being made. Decision support tools help decision makers improve their performance whilst decision-making tools automate the process, leaving a minimal role for the user. Tools that have been used to develop intelligent negotiation support systems include:

- Rule-based reasoning: where the knowledge of a specific legal domain is represented as a collection of rules of the form 'if then action/conclusion'.
- Case-based reasoning: uses previous experience to analyse or solve a new problem, explain why previous experiences are or are not similar to the present problem and adapts past solutions to meet the requirements.
- Machine learning: where the AI system attempts to learn new knowledge automatically.
- Neural network: consists of many self-adjusting processing elements cooperating in a densely interconnected network. Each processing element generates a single output signal that is transmitted to the other processing elements. The output signal of a processing element depends on the inputs to the processing element. Each input is gated by a weighting factor that determines the amount of influence the input will have on the output. The strength of the weighting factors is adjusted autonomously by the processing element as data is processed.

Traditionally, negotiation support systems have been template based, with little attention given to the role the system itself should play in negotiations and decision-making support. The primary role of these systems has been to demonstrate to users how close (or far) they are from a negotiated settlement. The systems do not specifically suggest solutions to users. However, decision support can be provided by informing users of the issues in disputes and the level of the disagreement.

Using Game Theory as a basis for providing intelligent negotiation support

Traditional negotiation decision support has focused on providing users with support on how to best obtain their goals. Such advice is often based on Nash's principles of optimal negotiation or bargaining.

Game theory, as opposed to behavioural and descriptive studies, provides formal and normative approaches to model bargaining.

Adjusted Winner and Smartsettle

Two widely known and used negotiation support systems are Adjusted Winner and Smartsettle. Both use game theoretic techniques to provide advice about what they claim are fair solutions. These algorithms are fair in the sense that each disputant's desire is equally met. They do not however meet concerns about justice. Both systems require users to rank and value each issue in the dispute by allocating the sum of one hundred points amongst all the issues. Given these numbers, game theoretic optimisation algorithms are then used to optimise, to an identical extent, each user's desires.

Adjusted Winner allocates divisible goods between two parties as fairly as possible. Adjusted Winner starts with the designation of the items in a dispute. If either party says an item is in the dispute, then it is added to the dispute list. The parties then indicate how much they value each item by distributing 100 points amongst themselves. This information, which may or may not be made public, becomes the basis for fairly dividing the goods and issues at a later stage. Once the points have privately been assigned by both parties, a arbitrator can use Adjusted Winner to allocate the items to each party and to determine which item (there will be at most one) may need to be divided.

Smartsettle is an interactive computer program developed to assist those involved in negotiating agreements among parties with conflicting objectives. It can be used during the negotiation process by opposing parties, or by a professional arbitrator. Smartsettle can help all parties to identify feasible alternatives that are preferred to each party's proposal based on information provided in confidence to the program. If such alternatives do not exist, the program can help the parties to develop counter proposals.

B.8 Arbitrator jury

The arbitrator jury is selected randomly after the Arbitration System has received the claim and accepted it.

Arbitrators work independently to validate real world information pertaining to transaction disputes and then cast votes on a mediated outcome.

Arbitrators receive case data and cast their votes using the arbitrator client. They are not able to communicate with one another and are unaware of how many other arbitrators are involved in resolving a dispute.

The number of arbitrators is dependent on the case and the dispute stage.

B.9 Merchant rolling reserve

The rolling reserve is a share of a merchant's transactions that is temporarily set aside to cover potential business risks, such as when a merchant loses a mediated dispute and must compensate the consumer. Rolling reserve funds are denominated in COTI coins and automatically accumulate in the merchant's account for a defined term.

The Arbitration Service creates a rolling reserve for each merchant to cover possible claims and a system-wide Reserve Credit Fund (RCF) to guarantee it. Both funds are maintained in COTI's native currency. The required size of a merchant's rolling reserve is calculated based on the merchant's Trust Score.

The rolling reserve is used when a merchant has lost a mediated dispute and is required to compensate the consumer. Merchants that do not meet the rolling reserve requirements will forfeit their ability to sell goods and services within the COTI network.

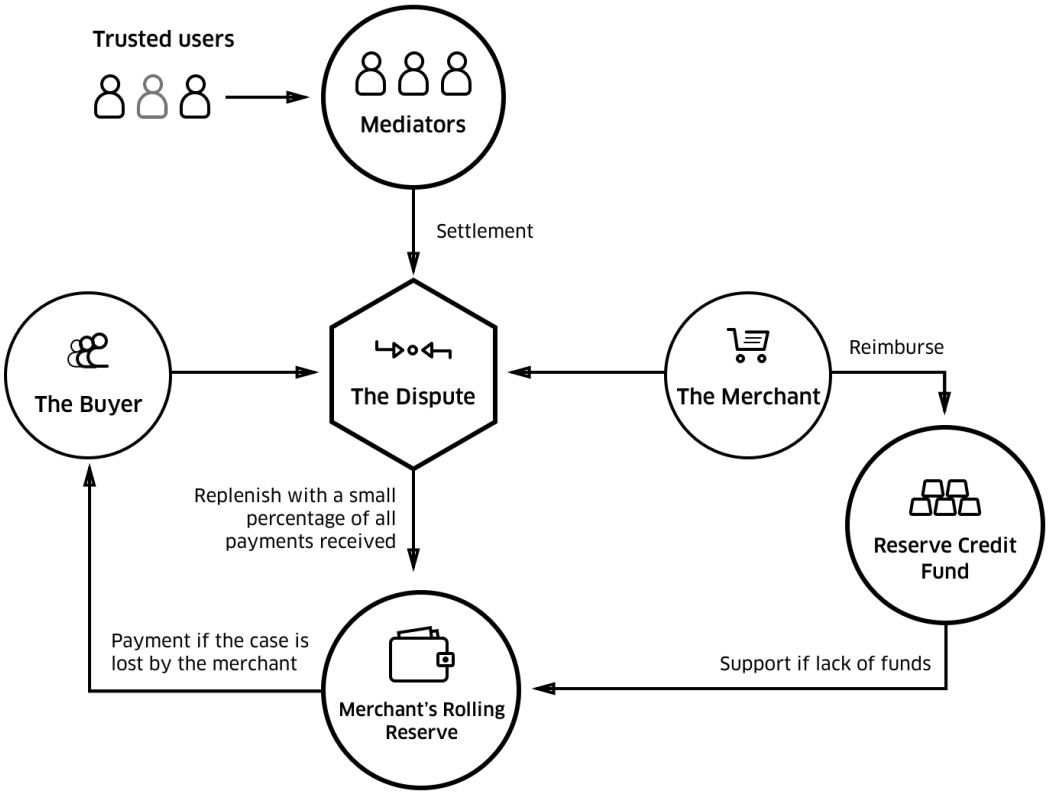
B.10 Arbitration rules

Trade dispute

There are two possible outcomes to a dispute. If the merchant wins, no additional transactions are needed. If the merchant loses the case, the Arbitration System creates a transaction that transfers money from the merchant's rolling reserve to the customer's account. If the merchant's rolling reserve is not sufficient, the Reserve Credit Fund (RCF) will be used, and the merchant is obliged to reimburse the RCF. In the COTI Arbitration System, the merchant has a fixed period in which he/she may pay voluntarily or make an appeal.

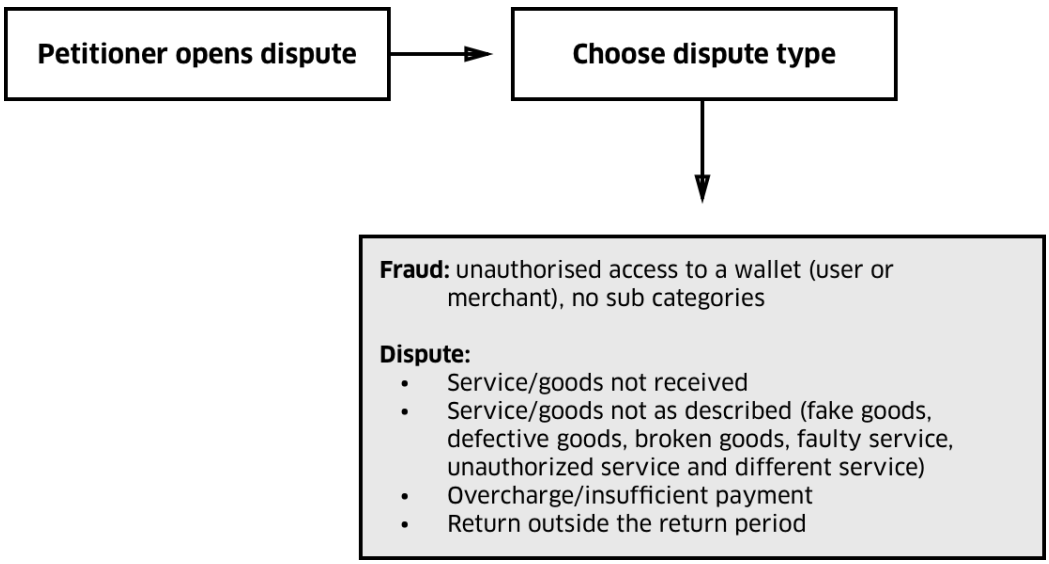
If the customer is the winning party, he/she is remunerated using funds from the merchant's rolling reserve. In case of insufficient rolling reserve funds, compensation is remitted from the RCF. When the merchant is the winning party, on the other hand, no further action is needed.

Figure 43: Pictographic summary of how COTI's trade dispute process works.

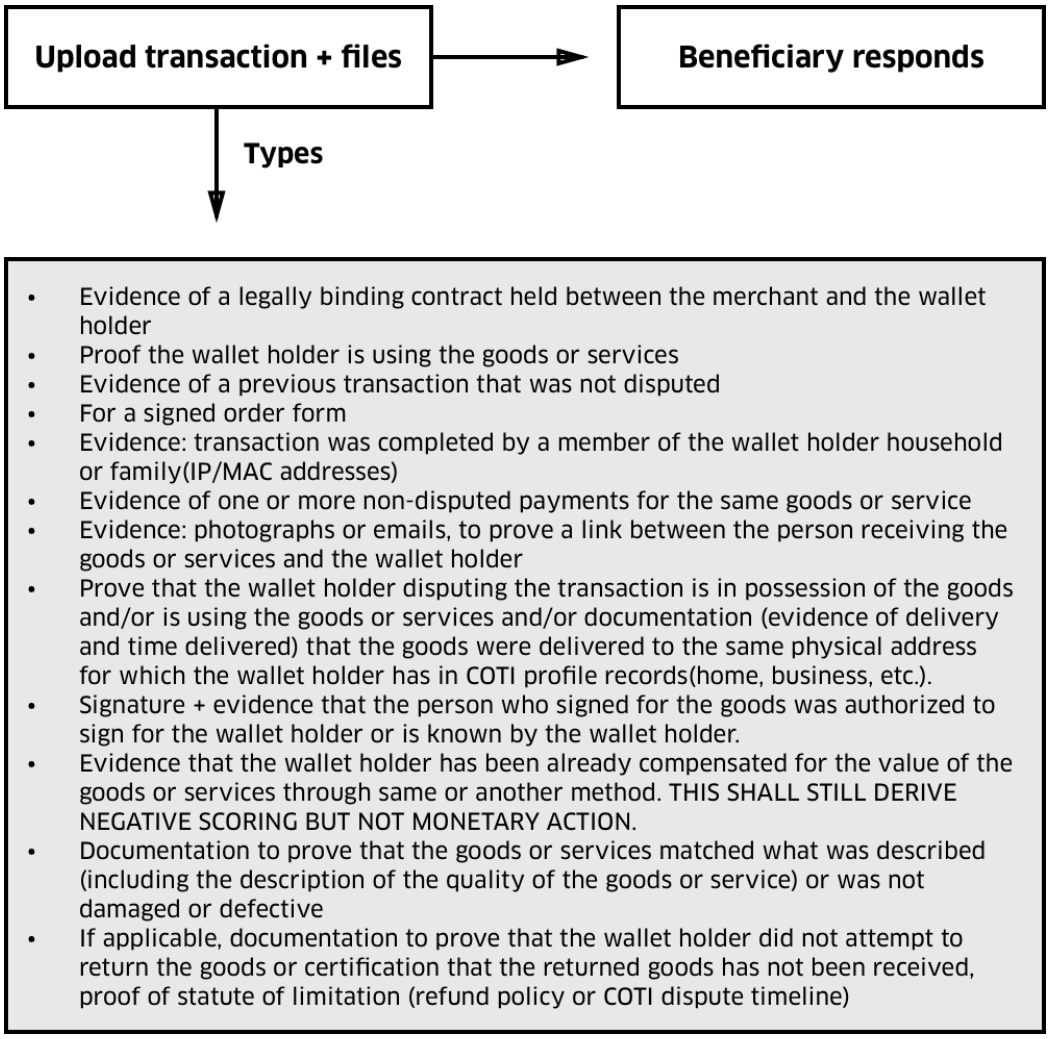


COTI arbitration flow

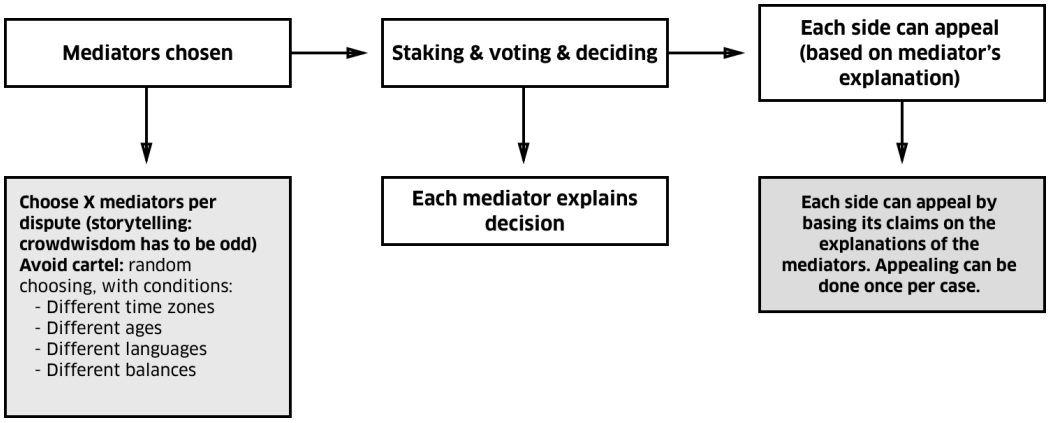
1. **Initiating a dispute:**



2. **Trying to resolve the dispute without arbitration:**



3. **Arbitration - selecting a jury:**



B.11 Fees

Arbitration Initiating Cost

A fee payable by the plaintiff upon complying. This fee is intended to decrease the demand for arbitration and avoid barratry. This fee is refundable if the plaintiff won the case.

Dispute resolution fee
The fee is dependent on the type of case and the stage of the dispute resolution and is paid by the party that lost the case.

Arbitrator fee
The fee that COTI arbitrators earn for participating in the dispute resolution process.

This fee is dependent on the case type, the stage of the dispute resolution, and the arbitrator's stake held in COTI coins.

Merchant rolling reserve fee Rolling reserve requirements are calculated based on the merchant's turnover and Trust Score, which is a powerful incentive for a merchant to maintain a high Trust Score.

Because of the design of the COTI payments network, Arbitration System and decentralisation, the rolling reserve requirements are substantially lower than existing payments networks. Every merchant transaction incurs a rolling reserve fee that is reserved for a a predefined arbitration time window. When the rolling reserve term has ended, funds are released back to the merchant's account.

B.12 Jurisdictions

The COTI Arbitration System resolves disputes between parties who voluntary agree to the mediated outcome. The COTI Arbitration System cannot deal with potential criminal cases, or substitute public law enforcement.

In general, any jurisdictional court will opine that it cannot accept a situation in which an international corporation focusing on the provision of its services to the local market can prevent clients from accessing other jurisdictional courts and adjudicating their dispute in accordance with local laws.

It can be stated, that any of COTI's operations should expect to be subject to the jurisdiction of local courts. In addition, it can also be stated that the mere fact that most, or all of the activities of such multinationals, is performed via the internet does not provide them with immunity from being taken to court and subject to local laws.

To this end, by offering products and services online, an online merchant may be subject to the jurisdiction of local courts, regardless of the merchant's attempt, via its terms and conditions, to subject its clients to the jurisdiction of foreign courts, or the COTI arbitration process.

It follows that any e-commerce merchant that has local clients, or is operational in any other form, or manner in any given jurisdiction should carefully review the scope of legal exposure it may have as a result of its activities and clients, and seek legal advice so as to limit such exposure by properly structuring its business.

B.13 Policies

Collusion prevention
Due to its distributed nature, the Arbitration System needs to take into account the possibility of collusion, either between arbitrators, or between arbitrators and one of the parties in a disputed transaction. This collusion risk is mitigated by a random selection of arbitrators to the jury. If any arbitrators are found to have engaged in any form of collusion, they will be severely penalised.

Privacy
Prior to distributing case data, COTI takes measures to ensure that only the data that aids directly in dispute resolution is disclosed. By default, COTI removes personal identifying information from all data submissions. If the parties to a dispute so choose, they can elect to forgo privacy for the sake of providing more detailed data. During the arbitrator registration process, all arbitrators are required to read and accept the arbitrator privacy policy, any violation of which will result in an expulsion from the COTI network.

B.14 Dispute attestation

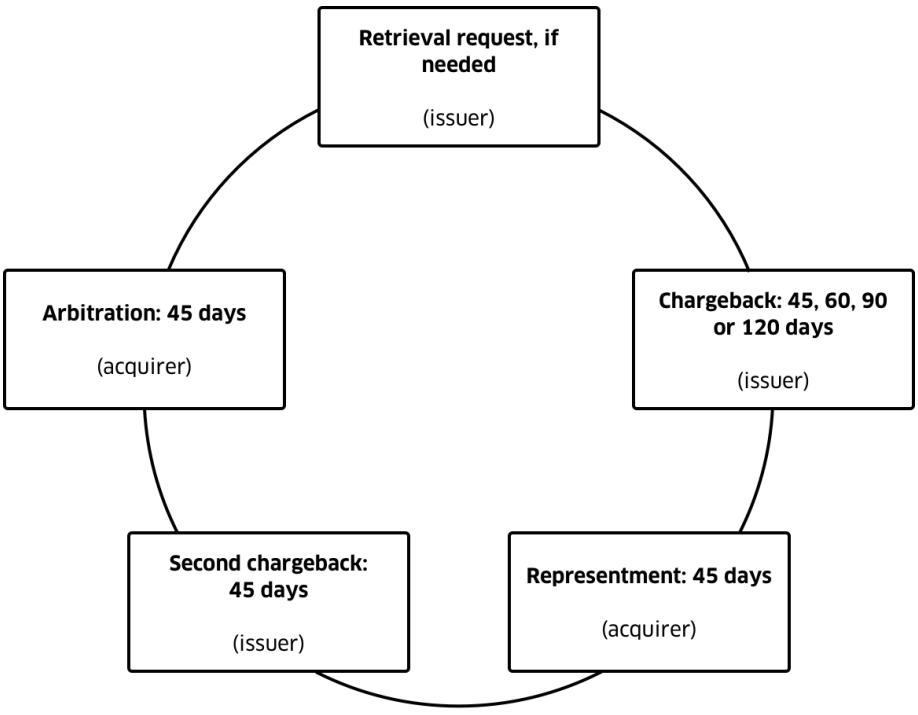
Possible questions for opening a dispute:

- Which transactions are you disputing?
- What is the reason for your dispute?
- When did you cancel the transaction(s)?
- Did you engage in the transaction(s), or receive any goods or benefits as a result of it?
- Have you attempted to contact the merchant?

What documentation is needed?
One of the following items is needed:

- Dispute Reason
- Receipt(s)
- Confirmations (emails, faxes, etc.)
- Proof of shipment or return
- Any other document that supports your case

Figure 44: **Overview of the dispute cycle**



B.15 Dispute influence on Trust Scores

Losing a arbitration case will be reflected in the Trust Score of consumers and merchants. The penalty only applies if X arbitrations are lost within XX transactions of one another and increases in severity as the frequency of lost arbitrations increases. Penalties will apply to both parties to prevent a user from unnecessarily lodging disputes.

Figure 45: Disputes influencing the Trust Score