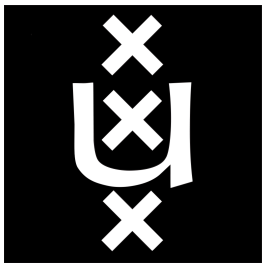

University of Amsterdam

Master Thesis

Software Engineering

A Realistic Approach to Trustless Asset Transport

Student Name:	Dylan Bartels
Student Number:	10607072
Supervisor:	Jaap van Ginkel
Host Organization:	CargoLedger

June 28, 2018

Abstract

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Contents

1 Introduction

The gig economy is in full effect, individual actors get payed for the execution of short term contracts and centralized companies intermediate in the supply and demand of this labour. With the intermediation of these parties the companies profit of margins and deny individuals full ownership of the value of the produced labour. Recent advancements in peer-to-peer technologies and decentralized posibilities interrest the academic domain if there are alternative options to shift towards decentralized solutions in the logistics domain.

1.1 Initial Study

Recent successful technical innovations have been due to a shift from centralization to peer-to-peer services, examples of these are Uber, Airbnb and Kickstarter. In the domain of supply chain logistics this innovation has been lacking. O. Gallay et al. [2] have proposed a peer-to-peer framework supporting interoperability between different actors in the logistics chains. This research lacks insight related to trust, network and technical implementation but establishes interest in the domain regarding implementation.

According to N. Hackius et al. [3] surveys in the logistics domain show that there is a clear demand on what blockchain technology can realistically do for the domain.

With the recent progression in the domain of trustless value transference, research towards the applicability of this in the supply chain offers relevancy. In [5] M. Klems et al. have formulated possible implementation of trustless intermediation in blockchain based decentralized service marketplaces. The research topic arises if this or other intermediation solutions can also be applied to peer-to-peer logistics marketplaces and to which degree will there be a custodian in the process due to it including transferring a physical asset.

1.2 Problem Statement

The problem which will be explored in this study is the possiblity of creating a peer-to-peer decentralized transport system where reputation is not a neccesity. Currently the transport domain operates around centralized reputation systems, whereby the companies with aggregated reputation and trust offer the service and carry responsibility for conflict resolution.

However reputation loss might not be the only incentive available to achieve transport. In chapter 4 an alternative incentive construction will be demonstrated which aims to achieve decentralization, reduction on trusting reputation systems and every actor being able to fullfill every role in transport. The setup uses trustless escrow to lock the transport actor into not behaving hostile due to possible punishment. Deviation of rational behaviour would result in loss of value to counteract the inavalibility of current applied reputation loss punishment.

1.2.1 Research Questions

Main research question:

- What are the specific problems and characteristics of a trustless decentralized peer-to-peer marketplace for transportation contracts?

Subquestions:

For the following subquestions marketplace is defined as a trustless decentralized peer-to-peer marketplace for transportation of goods.

- Can trustless intermediation exists on this marketplace without a custodian for dispute prevention and resolution?
- What level of anonymity is possible on this marketplace?

1.2.2 Solution Outline

Our solution uses a digital representation of the transport contract which includes the begin address, end address and the end address public key. This contracts will be called the asset and ownership on this contract will be tied to the owner of the physical asset which is intended to be transported with the process. When the transport actor is custodian of transporting the asset an equivalent value or more will be put in escrow which can be unlocked by the endpoint actor. The setup incentivezes for the transport actor to move the asset to the endpoint else losing the escrow equivalent value.

We chose to implement the escrow on the Bitcoin network due to it being well tested and offering a baseline environment in the decentralized domain.

1.2.3 Research Method

The study will apply the action research methodology research method. Action research can be defined as an approach in which the action researcher and a client collaborate in the diagnosis of the problem and in the development of a solution based on the diagnosis. With this method a prototype of the marketplace and transport intermediation solution will be build in collaboration with Cargoledger. The methodology has the downside that biases might occur towards the chosen solution due to also being responsible for the development.

1.3 Related Work

1.3.1 Building Trust in Decentralized Peer-to-Peer Electronic Communities [10]

Summary: Many players in electronic markets have to cope with much higher amount of uncertainty as to quality and reliability of the products they buy and the information they obtain from other peers in the respective online business communities. One way to address this uncertainty problem is to use information such as feedbacks about past experiences to help making recommendation and judgment on product quality and information reliability. This paper presents PeerTrust, a simple yet effective reputation-based trust mechanism for quantifying and comparing the trustworthiness of peers in a decentralized peer-to-peer electronic marketplace. There are three main contributions in this paper. First, we argue that the trust models based solely on feedbacks from other peers in the community is inaccurate and ineffective. We introduce three basic trust parameters in computing trust within an electronic community. In addition to feedbacks in terms of amount of satisfaction, we incorporate the feedback context such as the total number of transactions and the credibility of the feedback sources into the PeerTrust model for evaluating the trustworthiness of peers. Second, we develop a basic trust metric that combines the three critical parameters to compare and quantify the trustworthiness of peers. Third, we present a concrete method to validate the proposed trust model and report the set of initial experiments, showing the feasibility, costs, and benefits of our approach.

Differ from my approach: Xiong & Liu, 2003 focus on the wide domain of trust in decentralized peer-to-peer communities. With my specific case regarding a marketplace with transport of goods trust between the peers is an important aspect but is more contained in a specific domain than Xiong & Liu.
Obtained result: Xiong & Liu, 2003 present PeerTrust a trust mechanism for building trust in peer-to-peer electronic communities. They identified three important trust parameters, these are: amount of satisfaction, number of interactions and balance factor of trust. They put the results into experiments which demonstrated the effectiveness, costs, and benefits of the approach.
Remaining open questions: Looking at ways to make the approach more robust against malicious behaviors, such as collusions among peers. Combining trust management with intrusion detection to address concerns of sudden and malicious attacks. How to uniquely identify peers over time and associate their histories with them

1.3.2 The challenge of decentralized marketplaces [9]

Summary: Online trust systems are playing an important role in to-days world and face various challenges in building them. Billions of dollars of products and services are traded through elec- tronic commerce, files are shared among large peer-to-peer networks and smart contracts can potentially replace pa- per contracts with digital contracts. These systems rely on trust mechanisms in peer-to-peer networks like reputation systems or a trustless public ledger. In most cases, reputa- tion systems are build to determine the trustworthiness of users and to provide incentives for users to make a fair con- tribution to the peer-to-peer network. The main challenges are how to set up a good trust system, how to deal with se- curity issues and how to deal with strategic users trying to cheat on the system. The Sybil attack, the most important attack on reputation systems is discussed. At last match making in two sided markets and the strategy proofness of these markets are discussed.
Differ from my approach: Very similair by giving a rundown of all the research done towards trust enforcements in p2p file sharing, decentralized markets and sybil attacks.
Obtained result: B van Ijzendoorn gives a summary of academical research on decentralization, Sybil attacks, trust and peer-to-peer in relation to marketplaces.
Remaining open questions: Not applied.

1.3.3 A Peer-To-Peer Platform for Decentralized Logistics [2]

Summary: We introduce a novel platform for decentralized logistics, the aim of which is to magnify and accelerate the impact o ered by the integration of the most recent advances in Information and Communication Technologies (ICTs) to multi-modal freight operations. The essence of our peer-to-peer (P2P) framework distributes the management of the logistics operations to the multiple actors according to their available computational resources. As a result, this new approach prevents the dominant players from capturing the market, ensures equal opportunities for di erent size actors, and avoids vendor lock-in. The latest ICTs such as Industrial Data Space (IDS), Blockchain, and Internet-of-Things (IoT) are used as basic building blocks which, together, enable the creation of a trusted and inte- grated platform to manage logistics operations in a fully decentralized way. While IDS technology allows for secured data exchange between the di erent parties in the logistics chain, Blockchain technology handles

transaction history and agreements between parties in a decentralized way. IoT enables the gathering of real-time data over the logistics network, which can be securely exchanged between the di erent parties and used for managing the decision-making related to the control of the freight transportation activities. The practicability and the potential of the proposed platform is demonstrated with two use cases, involving various actors in the logistics chains.
Differ from my approach: It is an academic research which originated from the logistics domain and aimed at solving the contradiction between interoperability and data sovereignty.
Obtained result: High level decentralized logistics system architecture with data flows. Two initial use cases.
Remaining open questions: Development of business models in parallel.

2 Background

2.1 Bitcoin

Introduced by the anonymous Satoshi Nakomoto in 2008, Bitcoin (BTC) is a decentralized peer-to-peer currency system. Bitcoin allows digital payments without going through a financial institution and solves the double spend problem by hashing timestamped transactions into an continues chain of hash-based Proof-of-Work (PoW) [6]. Payment are possible by creating transactions, signing them and sending them to Bitcoin addresses. The user has a public/private keypair whereby addresses are a mapping function of the public key and the private key can sign transactions. Creating of transactions is only possible if they have been send to one of the user owned addresses and are then called unspent transactions (UTXO).

The broadcasted transactions are send to the memorypool and included in blocks once the network has formed consensus on the correct order of transactions. To generate a block the miners have to find a nonce value, peers than include it in a block which allows anybody to verify the PoW. Miners get rewarded upon generating a block thus incentivizing to support the network with computational power. The generated computational power of the network, also expressed in hash rate or hash power, protects the intregrity of the PoW chain. For a user the definition of owning a bitcoin is the right to sign a UTXO with your keypair.

The PoW mechanism solves the double spend problem by garanteeing that the transaction is not spend twice when it is included in a block. For a malicious actor to double spend a BTC without detection they would have to recompute all previous blocks, so as long as the honest peers in the network exceed the malicious the intregrity of the work is garanteed [6].

The average block time of the bitcoin network is 10 minutes, to garantee that the double spend would not occur on average the receiver of the transaction would have to wait 10 minutes minus the time of last found block. Fast transactions (i.e. in the order of seconds) are not reliable because low cost attacks can be mounted effectively to spoof a transaction [4].

2.1.1 Trustless

Markus Klems et al. define trustless and trustless intermediation as follows:

> *A system property which guarantees rules of interaction that are known to and agree upon by all participants of the system, and which cannot be unilaterally changed. These guarantees are enforced through, what we call trustless intermediation, a set of mechanisms for decentralizing the enforcement of rules in a system, thereby removing the need for and existence of trusted intermediaries. [5]*

Bitcoin can be defined as a trustless system because once an UTXO is signed and broadcasted to the mempool with a high enough transaction fee it is garanteed that it will be put inside a block. No intermediation takes place for this mechanism to occur. The mechanism cannot be changed *easily*, only if a hard forking is proposed including a code change. However this chain split would not count as bitcoin unless it is backed by the majority of hash rate thus being classified as longest chain [6].

Extrapolating the definition to the logistics domain means that trustless logistics can be defined as an logistic contract which has predefined unchangeable rules and no need for trusted intermediaries.

2.1.2 Script

Bitcoin uses a stack-based scripting system for transactions which is intentionally not Turing-complete. One possible output script is the multisigniture script which allows any of N signatures out of M to spend the UTXO available in the generated multisignature address. Bitcoin provides the OP_CHECKMULTISIG operators for multisig verification which make it possible to verify a generated transaction for the content before it is broadcasted.

2.2 Smart Contracts

Smart contracts are digital representations of a contracts which will be activated once certain input activates parameters. The contracts promise to: enforce contracts automatically Take out the middle man in contract construction, execution and enforcement A normal contract smart contracts are very difficult to implement well. They all trust on some oracle input which has to be correct for contracts to behave. but how can you garantee the input is correct? Conesnsus on the correctness of data which is stored depends on the correctness of input. If the oracles who register the data are in full control of input they are lone ruler of correctness disregarding the consensus.

2.3 Decentralized Marketplace

Markus Klems et al. define centralized marketplaces[5] as providing mechanisms to facilitate efficient spot trades between numbers of sellers and buyers by providing match-making and payment transaction processes that are accompanied by trust-building mechanisms, most importantly, reputation and dispute resolution systems. Some examples of centralized marketplaces for logistics are Postmates for deliveries, Uber for the transport of people and Uship for shipping of goods. These peer-to-peer marketplaces facilitate the intermediation process of transport contracts.

Decentralized marketplaces facilitate the same spot trades without a central provider and intermediaries. The upside of such a mechanism is the lowering of the entry barrier [1], no intermediation fees [11], increasing sensorship resistance [7], and improving privacy [8].

3 Test Setup

As a new idea, this study introduces the concept of trustless transport which replaces the need of centralized intermediation of supply and demand. We propose a mechanism which punishes hostile actors automatically resulting in no conflict resolution required from central entities.

In our scenario seen at figure 3.1, we assume that the service consumer A wants to send an physical asset to the endpoint actor B, whereby asset payment between them already took place. The third and last actor is the service provider C who will execute the transport. Let C, A and B all have an ECDSA key pair $\{PubK_n, PrivK_n\}$ and address Adr_n.

The scenario starts of with A creating a request for transport minimally containing B public key, B location and A location $\{PubK_b, Loc_b, Loc_a\}$. This request is accepted by C which then signs $\{PubK_c, PrivK_c\}$ a UTXO, denoted by *tx1*, containing the assets equivalent cost or more to the 2/2 multisignature address of $\{PubK_b, PubK_c\}$ denoted by $MSig_{bc}^{n/2}$.

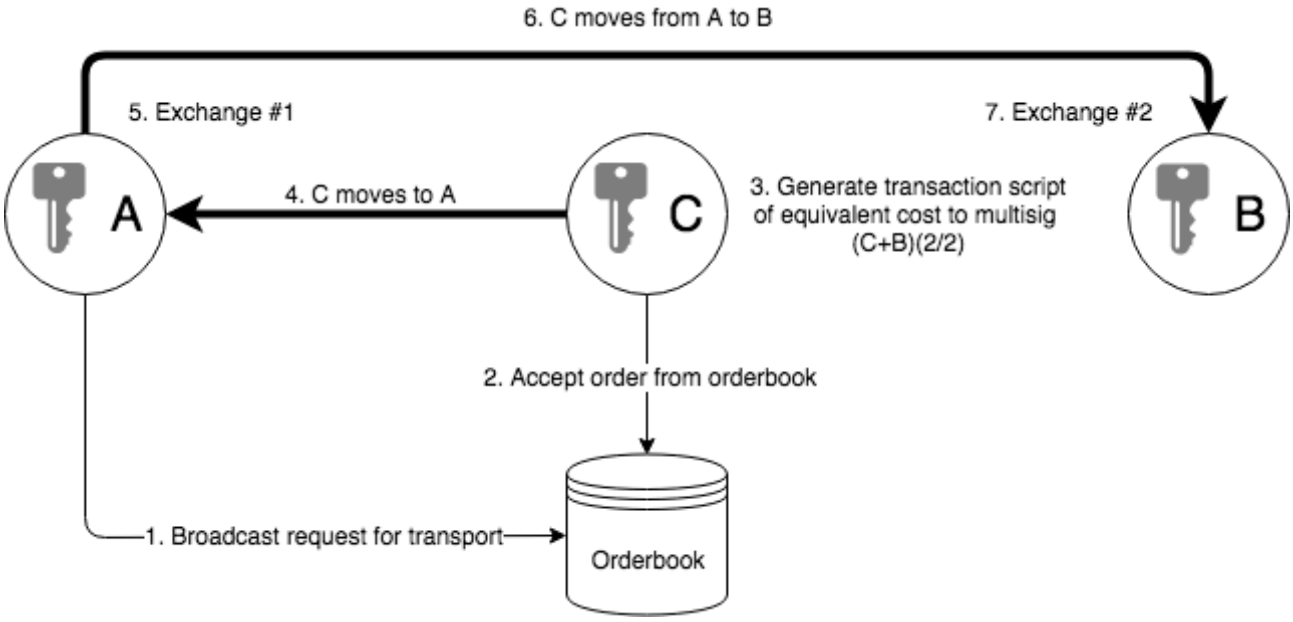

Figure 3.1: Overview test scenario

Upon accepting C moves to the physical location of A bringing transaction script *tx1*. As illustrated with figure 3.2, upon C arriving at A the first exchange can take place.

1. C innitiates the exchange by giving A the transaction script *tx1*
2. A generates and signs $\{PubKa,\ PrivKa\}$ UTXO *tx2* containing the transporting cost of the physical asset to $MSig_{bc}$
3. A can now broadcast *tx1*, *tx2*
4. A broadcasts the ownership change of the digital asset from $A \rightarrow C$
5. Wait block confirmation containing *tx1* and *tx2*
6. Exchange the physical asset from $A \rightarrow C$.

Before *tx1* and *tx2* get broadcasted $A \vee C$ can individually verify if the signed transactions actually contain what they should.

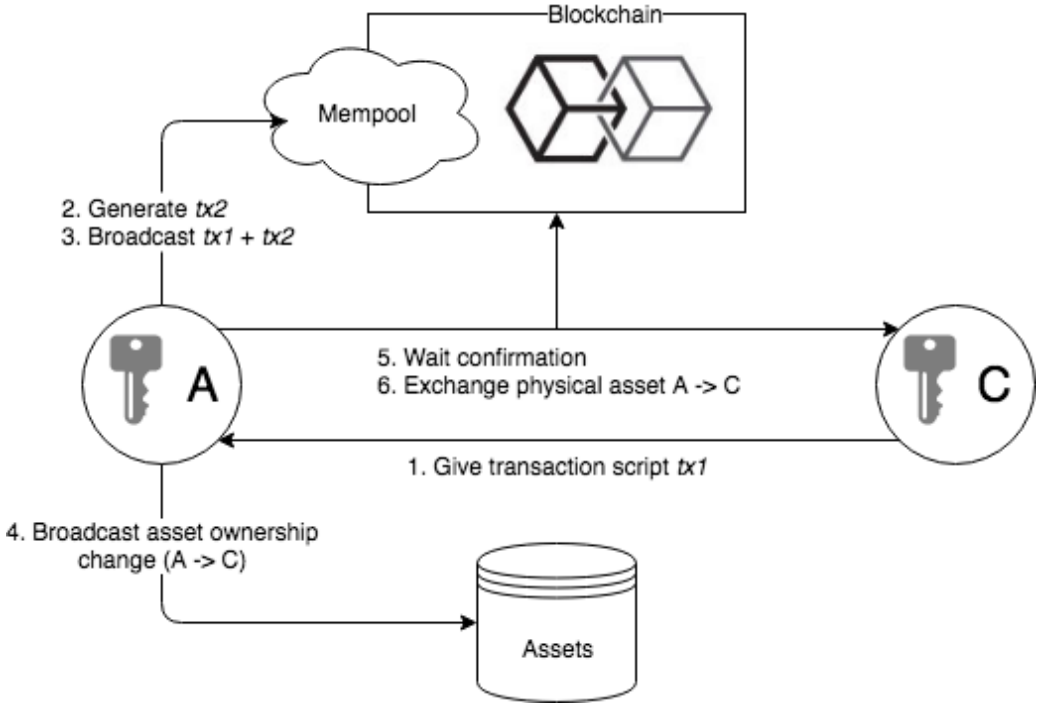

Figure 3.2: Exchange between A and C

When C recieves the asset he is the custodian and will move the asset to the endpoint Loc_b. As illustarted with figure 3.3, upon C arriving at Loc_b the physical asset dropoff exchange takes place which consist out of the following steps:

1. B signs $\{PubK_b, PrivK_b\} \rightarrow MSig_{bc}^{1/2}$ the two previously send UTXO's $\{tx1,\ tx2\}$ containing $\{equivalent\ cost\ +\ transport\ cost\}$ to the address of C Adr_c and gives this transaction script to C.

2. Exchange physical asset $C \rightarrow B$

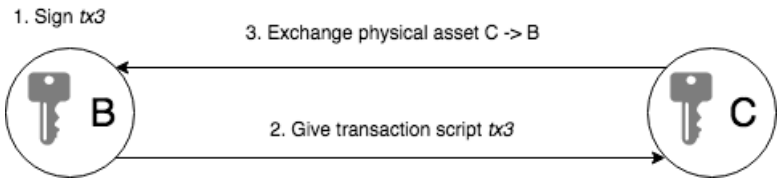

Figure 3.3: Exchange between C and B

At the end of the second exchange actor C now owns the transaction script containing $\{ec{+}tc\}$ of $MSig_{bc}^{1/2}$ to be send to Adr_c and can sign the transaction with his own keypair whenever he wants to redeem the funds $\{PubK_c, PrivK_c\} \rightarrow MSig_{bc}^{2/2}$.

4 Result

In this chapter we describe OpenLogistics, the created decentralized marketplace prototype for transport contracts. OpenLogistics uses the bitcoin mainet for the multisignature mechanism and bigchaindb to create orders and aggregate them in a orderbook.

To analyze the results the whole testsetup will be devided into different phases, at every phase the possible actor actions will be evaluated. The phases to be analyzed are the following:

1. Setup
2. Pickup
3. Dropoff
4. Redeeming

What is really being exchanged is the right to the transaction id by being possible to acces it with the private key. What the transaction id represent in the orderbook is proof that asset ownership is possible to acces. The exchange of owner innitiates the escrow balance which is equavalent to the asset value.

4.1 Transparancy

A example of a mainnet testrun can be seen locally, if copy of blockchain is present, or through commonly used block explorers [add as footnote] with the following addresses:

1. Actor A: 17F4ZhEJp83qqEG1S6z8YcPbWW7AdqbkZ3
2. Actor B: 19exDB5Fb2gQAv7k2dH93WbLga1ZUNz9mh
3. Actor C: 1EY38FGwuSg3uRzetBwYqYh9jjbX55fHsL
4. Multisig (B+C): 3MiFyavsRpMZBzfxFk94WdeZUnbQP1hdDy

Actor B actually has no UTXO on his address, this is due only using his keypair to sign the transactions. The function Actor B actually fulfills is being the oracle for conflict resolution upon dropoff exchange by signing or not.

The asset transferring taking place can also be seen by exploring bigchaindb addresses.

The aggregation of the movement of assets between addresses can be used for a reputation system.

5 Conclusion

Attack vectors: Eating the difference in equivalance cost if the reputation cost of the starting actor is benefitial to the actor eating the costs blockchain is the right to sign a UTXO, the right is valued due to being censorship resistant. The capacity to make it censorship resistant cost a lot of energy. From an economic perspective centralization is very efficient due to not needing to form consensus which cost a lot of energy. Traditional reputation systems in place form all the "aanspraakelijkheid" the smart logistics contract resembles. They do this quite well, if bad events occur when tnt post is custodian of your asset than resolution is often very dynamic. This would be hard to capture in similair smart contract construction as OpenLogistics[1]. Conflict resolution is currently a loss for all parties when a package is lost, these kind of conflict could beter be immediated by current traditional logistics caretakers. It's only economically viable to spend such an amount of energy to defeat the possible censorship that it becomes viable. Zero confirmation: Zero confirmations are currently not safe to be used in production setups, the attacker could easily write a script broadcasting the same UTXO moments later. This double spend attack will remain harmfull in the current testsetup. Solutions are available, bitcoincash[cite safe accept 0-conf] accepting zero-conf safely, microsoft currently accepts this payment since declining to accept bitcoin for the mentioned reason. Since bitcoincash is a fork of bitcoin the testsetup could easily be alternated to work on this network. Another possiblity to counteract the double spend attack would be to use the bitcoin lightning network which is a second layer solution which enables instant transactions. Given that blockchain mainchain does not scale and remain decentralized this alternative would provide more promise. RBF (replace-by-fee): no rbf: 51 attack or Finney attack(https://bitcoin.stackexchange.com/questions/4942/what-is-a-finney-attack) A network like ethereum has a average block time of 15 seconds and with a is quite safe after 8 confirmations.

The bitcoin protocol does not provide any guarantee at all about zero-conf transactions. It provides probabilistic guarantees for n-conf transactions, but only when n is 1 or more. Specifically, as Satoshi proved, the probability of an n-conf transaction being reversed and double-spent decays very quickly with n, so that for n = 6 it can be considered impossible.

Attempts to get zero-conf payments to work, e.g. by inspecting the mempools of volunteer relay nodes, are attempts to solve the double-spending problem without the blockchain, the miners, and proof-of-work. People tried to do that for 25 years with no success.

You can get a solution for the 0-conf double-spend problem that "works" only in the hacker's sense of the word, not in the engineer's sense. That level of security is OK for bittorrent or Tor – but not for a payment system.

The only change that would actually improve bitcoin's dismal usability (and make it competitive with other altcoins) is a reduction of the block interval from 10 minutes to under a minute. Then it would still be too slow to compete with credit and debit cards for walk-in stores and restaurants, but may be good enough for e-commerce.

By the way, the payment channels that are supposed to be the building block of the Lightning Network use zero-conf transactions that are kept on file for months before being sent to the miners. Thus payment channels too are secure only in the hackers' sense of the term – that is, not really secure.

6 Discussion

blockchain is nothing other than the right to sign a UTXO, need to depend on oracles to verify the data which it represents.

References

[1] Liran Einav, Chiara Farronato, and Jonathan Levin. Peer-to-peer markets. *Annual Review of Economics*, 8:615–635, 2016.

[2] Olivier Gallay, Kari Korpela, Niemi Tapio, and Jukka K. Nurminen. A peer-to-peer platform for decentralized logistics. In *Digitalization in Supply Chain Management and Logistics*, oct 2017. http://tubdok.tub.tuhh.de/handle/11420/1476; Proceedings of the Hamburg International Conference of Logistics (HICL).

[3] Niels Hackius and Moritz Petersen. Blockchain in logistics and supply chain: trick or treat? In *Proceedings of the Hamburg International Conference of Logistics (HICL)*, pages 3–18. epubli, 2017.

[4] Ghassan Karame, Elli Androulaki, and Srdjan Capkun. Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin. *IACR Cryptology ePrint Archive*, 2012(248), 2012.

[5] Markus Klems, Jacob Eberhardt, Stefan Tai, Steffen Härtlein, Simon Buchholz, and Ahmed Tidjani. *Trustless Intermediation in Blockchain-Based Decentralized Service Marketplaces*, pages 731–739. Springer International Publishing, Cham, 2017.

[6] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[7] J. Winter M.J.G. Olsthoorn. Decentral market: self-regulating electronic market. *DelftUniversity of Technology*, 2016.

[8] Kyle Soska, Albert Kwon, Nicolas Christin, and Srinivas Devadas. Beaver: A decentralized anonymous marketplace with secure reputation. *IACR Cryptology ePrint Archive*, 2016:464, 2016.

[9] Bas van IJzendoorn. The challenge of decentralized marketplaces. *CoRR*, abs/1703.05713, 2017.

[10] Li Xiong and Ling Liu. Building trust in decentralized peer-to-peer electronic communities. In *In The 5th International Conference on Electronic Commerce Research. (ICECR*, 2002.

[11] Dionysis S. Zindros. Trust in decentralized anonymous marketplaces, 2015.