DECENTRALIZED REPUTATION MODEL AND GENERAL TRUST FRAMEWORK BASED ON BLOCKCHAIN & SMARTCONTRACTS

Dissertation in partial fulfillment of the requirements for the degree of

MASTER PROGRAMME IN COMPUTER SCIENCE

UPPSALA
UNIVERSITET

Uppsala University
Department of Information Technology

SUJATA TAMANG

May 22, 2018

BLOCKCHAIN BASED DECENTRALIZED REPUTATION MODEL AND GENERAL TRUST FRAMEWORK

Dissertation in partial fulfillment of the requirements for the degree of

MASTER PROGRAMME IN COMPUTER SCIENCE

Uppsala University
Department of Information Technology

Approved by

Supervisor,	Jonatan Bergquist
Reviewer,	Björn Victor
Examiner,	Prof. fName lName

May 22, 2018

Abstract

Abstract here

Acknowledgements

Acknowledgements here

Table of Contents

List of Tables

List of Figures

List of Abbreviations

1 Introduction

1.1 Definition

1.1.1 Trust and Reputation

Trust encompasses a broad spectrum of domains and is context dependent. Therefore, its definition varies based on context and discipline and as such lacks collective consensus among researchers [1] [2]. Using the classification from McKnight et al., 1996, Trust can be either Personal/Interpersonal, Dispositional or Impersonal/Structural. Personal trust is when one person trusts another specific person, persons, or things in a particular situation. Interpersonal trust involves more than one trusting entities. i.e., two or more people (or groups) trust each other. Dispositional trust refers to a more general trust that is based on personality attribute of the trusting party. It can be seen as a sense of basic trust(attitude) and is cross-contextual. While the trust mentioned above are implicitly directed towards a person, Impersonal/structural trust is more likely to refer to an institutional structure such as a judiciary system.

Trust can be generally seen as an entity's reliance on another interacting entity to perform a specific set of the task given a specific situation. As pointed out by [3] "Trust is the subjective probability by which an agent assesses that other agent or group of agents will perform a particular action that is beneficial or at least not detrimental." For an entity, 'A' to trust another entity 'B' or to evaluate B's trustworthiness, the reputation of 'B' plays a central role. Broadly defined, Reputation is the perception of an individuals character or standing. Like Trust, reputation is context-dependent. e.g., Alice may be trusted to answer or use Linux questions efficiently but not Windows related questions. [4] A significant difference between trust and reputation is that the former takes the subjective measure as input whereas the latter takes an objective standard (e.g., transaction history, ratings) as an input to yield a resulting score that can aid in detecting reliability/trustworthiness of an entity. [5] [6]

The classification of trust and reputation measures based on [7]:

	Specific, vector-based	General, Synthesized
Subjective	Survey questionnaires	eBay, voting
Objective	Product tests	Synthesised general score from product tests, D&B rating

Individuals in online systems are identified by their online identities which can be anything and not necessarily linked to their real-world identities. Online identities play a crucial role in digital interaction and require unknown entities to trust each other based on the reputation system of the platform in use. As mentioned in [8], trust and reputation are soft security mechanisms where it is up to the participants rather than the software/system to maintain security. Unlike hard security mechanism such as access control, capabilities, authentication where a user can be allowed or rejected access to the resource. Reputation system aids in calculating the probability of success or risk of failure of a transaction between interacting parties.[9][10]

1.1.2 Blockchain

Blockchain can be defined as a distributed record of state changes that let anybody on the network audit state changes and prove with mathematical certainty that the transactions transpired according to the blockchain rules[1]. There exist several definitions of blockchain technology each specific to their closest use case. A formal standard definition of Blockchain is under development as ISO/TC 307. [2]

Vitalik Buterin, the founder of Ethereum, puts it this way:

> "A blockchain is a magic computer that anyone can upload programs to and leave the programs to self-execute, where the current and all previous states of every program are always publicly visible, and which carries a very strong cryptoeconomically secured guarantee that programs running on the chain will continue to execute in exactly the way that the blockchain protocol specifies." [3]

This definition provides a broad overview of what blockchain does. As a continually developing discipline, it keeps adapting to a new definition while maintaining the essence. The major innovation of blockchain as an architecture is distributed, decentralized trustless transactions[11]. It completely removed the need for an intermediary trusted third party by building trust in the system itself. One dimension of trust as mentioned by [12] is trust in data which is based on stored data's integrity. Trusting data ensures that the data is appropriate for use: accurate, precise, available, and uncorrupted[12]. Blockchain achieves this by use of cryptographic schemes(cite section) assuring tamper-resistant, fault-tolerance, zero-downtime characteristics[13].

[1] https://media.consensys.net/time-sure-does-fly-ed4518792679
[2] https://www.iso.org/committee/6266604/x/catalogue/p/0/u/1/w/0/d/0
[3] https://blog.ethereum.org/2015/04/13/visions-part-1-the-value-of-blockchain-technology

1.2 Motivation

Consider a simple scenario where Alice wants to buy a pair of headphones for which she browses a buy/sell platform. When she finds a relevant product on the platform published by Bob, unknown entity to Alice, the success or failure of the transaction is dependent on two factors that may or may not be transparent.
(i)Bob's reputation: Bob's reputation can be inferred from his history of transactions, ratings provided by previous buyers that have dealt with him, reputation system of the platform in use, the integrity of all these relevant data. (II) Platform's reputation: Reputation of the platform can also be inferred similarly based on the history of services it has been able to provide, a general perception in the community, etc. Here, the platform in use acts as the trusted third party that Alice must trust to present correctly computed, untampered data about Bob. The entity claiming to be Bob could be Eve who found a way to bypass the platform's security and inflate his reputation. Eve could delete the ad and associated account when the payment is complete, or she could gather Alice's details to misuse it later. Any malformed decision on the trustworthiness of an entity could be expensive and deal severe damage to the user.
Statistics suggest that online shopping is the most adapted online activity.[4] Reports by Experia [5] and Javelin [6] indicate that E-commerce fraud has risen to 30% in 2017 from 2016 while identity fraud victims have risen by 8% in 2017(16.7 million U.S victims). Additionally, reports on fake news [7] [8] that leads to spread of misinformation from malicious users or portals, attack on an existing system continues.
A recent report on Finland's data breach exposed [9] 130,000 users login details while Facebook has admitted to the compromise of 2.2 billion of its user's data[10]. While there are several security reasons that have led to attack at such scale. One major reason is the client-server architecture where everything is stored on a centralized server and data flows in and out from the same source. On the other hand, distributing information over a decentralized network would require a simultaneous attack to achieve the same effect, thereby increasing the difficulty level of attack. Similarly, Reputation models can help in measuring the reliability of interacting entities so that users can make an informed decision before participating in any transactions. Thus, a reputation system should be secure, robust, always available and aim for higher accuracy. The use of right reputation algorithms with Blockchain technology can help to ensure trustworthiness of online entities with correctness of data and a high degree of accuracy.

[4]https://www.experian.com/assets/decision.../reports/global-fraud-report-2018.pdf
[5]https://www.experian.com/blogs/ask-experian/the-state-of-online-shopping-fraud/
[6]https://www.javelinstrategy.com/press-release/identity-fraud-hits-all-t ime-high-167-million-us-victims-2017-according-new-javelin
[7]https://journalistsresource.org/studies/society/internet/fake-news-conspiracy-theories-journalism-research
[8]https://www.prnewswire.com/news-releases/84-percent-of-businesses-could-reduce-fraud-risk-if-certain-about-customers-identity-300587192.html
[9]https://thehackernews.com/2018/04/helsingin-uusyrityskeskus-hack.html
[10]https://thehackernews.com/2018/04/facebook-data-privacy.html

1.3 Purpose and research questions

The main goal of this thesis is to use blockchain technology and smart contracts to simulate an endorsement network where entities can endorse each other based on physical or digital acquaintance. The endorsement values will be quantified to infer reputation score and a trust value that can be used on any transaction network. The nodes and their relationship will be studied to identify honest or malicious participants. Generalization of this endorsement network to serve other use cases will also be discussed. The research questions that this thesis aims to address are :

1. How can graph theories and relevant reputation algorithms be used to model the interaction between entities and detect/identify honest and malicious nodes in the network? How can the interaction graph be modeled?
2. What are the requirements for storing trust values and linking them to associated identities stored off a blockchain network? How can a blockchain application be built to define a general trust framework for a transactional network? How could the overall system architecture look like?
3. How can the discussed endorsement network ensure trustworthiness while also preserving users anonymity and how can it be generalized to other transactional network or added on top of it to serve other use cases such as content filtering, E-Commerce etc?

1.4 Scope

This thesis work attempts to answer all the research questions mentioned in section 1.3.

To answer research question1, literature survey will be performed on existing reputation algorithms along with the presentation of background overview that will lead to graph simulation of endorsement network.

For research question2, interpretation and quantification of reputation scores and trust metrics will be manifested. Comparative analysis of on chain and off-chain storage requirements will be studied resulting in an overall design of endorsement system architecture.

For research question3, relevant use cases will be presented, and the network will be tested on with various predefined cases and attack models to see how well it behaves in a dynamic environment.

1.5 Structure of Report

This paper is structured as follows. In Chapter 2, a background on relevant concepts that will be needed to understand the following sections will be explained. Chapter 3 will explain the system requirements and design in detail. In chapter 4, the paper concludes by presenting the results and its evaluation.

1.5.1 acronyms

2 Literature Review

The earliest and most known internet reputation is that of eBay[1]. It uses a feedback based rating system where a user can rate a transaction along with some textual feedback. The range of values used being [1, 0, -1], positive, neutral and negative respectively. The final aggregated score is computed by subtracting the total of positive and negative ratings. This system [14] [15] could be judged as working based on the sales volume and the observation that more than half the buyers usually engage in providing feedback). However, this method fails to address issues such as Sybil attack, inactive participation(e.g., users fear retaliation from giving negative feedback). There are many none E-commerce online systems such as StackExchange [2], Yelp [3], Quora[4], Reddit [5] e.t.c, that make use of similar reputation mechanism to filter the participating users and avoid serving malicious participation. Most of the E-commerce systems employ a client-server architecture which lets a central entity in control of stored data. However, Peer-to-Peer(P2P) systems(cite) for applications such as file sharing, content delivery e.t.c. have studied and implemented reputation systems as well for detecting the quality of file/content as well the peers that upload them. [16]points out that a reputation system should be able to provide enough information to help infer the trustworthiness of participating users, encourage a user to be trustworthy and discourages the dishonest behavior.[17] is a reputation system that addresses these concerns. It introduces the role of insurers between interacting entities such that a user can ask to be insured for their transaction or insure someone else's transaction in exchange for a reward. The system relies on the insurer's capability of estimating failure probability.
[18] proposes a block-chain based reputation model that uses a blind signature(cited) to disconnect the link between customer and ratings. Doing so lets a customer freely rate/review the transaction without fear of retaliation. It is more customer-centric in the sense that it allows only a customer to rate the transaction. Thus, Sybil identities is not a concern here as a customer is allowed to make multiple identities and in fact, a unique identifier is recommended for a unique transaction. The obvious problem here is the unfair rating attack. A buyer can transact with a service provider provide negative feedback with an intention to damage the service provider's reputation despite their honest behavior in the network.

[1] https://www.ebay.com/
[2] https://stackexchange.com/
[3] https://www.yelp.com/
[4] https://www.quora.com/
[5] https://www.reddit.com/

2.1 Reputation Algorithms

The most known and widely used reputation algorithm in a P2P network is EigenTrust(cite). It recommends a method to aggregate local trust values of all peers. It has the notion of transitive trust and lets peers rate another peer as either positive or negative just like eBay does [-1,+1]. Based on the aggregated score, a user can decide if a peer can be considered trustworthy for a download source or not. Drawbacks (cite 2) of EigenTrust is that peers are likely to center around a static set of pre-trusted peers(cite). Pre-trusted peer is a notion of trust where first few peers that joined the network are assumed trustworthy by design. This notion is also used by Advogato's trust metric. There have been proposals to improve upon EigenTrust such as Incremental Trust, Honest Peer algorithm. There are two main methods in a reputation method system(cite): peer-based reputation system and file-based reputation systems. The former tries to rate a peer and associate a value with the peers that are actually participating in the network. The latter checks the integrity of a file that is being delivered/served on the network regardless of who(peer) owns or serves it. [AuthenticPeer++] is a trust management system for P2P networks that combines both of this reputation system. As such, it only allows trusted peers to rank the file after they download it and uses DHT-based structure to manage the integrity of file information.

3 Background

3.1 Graph properties

A graph, as the name suggests can be used to represent objects and their relationships graphically. A graph G is an ordered triple (V,E,φ_G) where V is a non empty set of vertices v, E is a non empty set of edges e that connects two vertices and $v \in V, e \in E$. φ_G is an incidence function that assigns pair of vertices to each edge of the graph G. φ_G(e) = uv represents that e is an edge that joins vertices u and v. Graph properties can be leveraged to serve as an interaction graph of network for reputation system. Each node on the network, v can represent individuals and the edges that connect the nodes can represent the relationships between those nodes. The edge can have varying weights to represent the strength of relationship between the nodes. [19]

3.1.1 Net flow Rate convergence

Net flow rate convergence can help to determine anomaly in the network. By looking at how fast the net flow converges to zero, it can detect unusual behaviour in the network. The flow in a network can be measured by looking at inflow and outflow edges and calculating their differences. Inflow edges are all incoming edges in the graph and outflow edges are all outgoing edges . (diagram) Net flow convergence rate is the rate at which the net flow converges to the global net flow which is zero. Depending upon how fast the net flow in a graph converges to zero, it can be useful to detect anomaly.(example diagram)

3.2 Cryptography

3.2.1 Basic Concepts

Cryptography offers algorithms to achieve confidentiality, integrity, authenticity, and non-repudiation. Confidentiality and integrity ensure that the information being communicated is not disclosed or has been modified to or by any unauthorized parties. The data is hidden or encrypted such that only the authorized parties can make sense out of it, i.e. decrypt using the previously agreed upon key.
Asymmetric key cryptography makes use of key pairs, private key, known only to the owner and public key, that can be publicly distributed. It ensures authenticity, a proof that sender is who he claims to be and non-repudiation, the sender cannot deny having

sent the message. Public key verifies the holder of the private key and encryption of the message. That paired private key can only decrypt this encrypted message. One of the significant application of public key cryptography is Digital Signatures, described in more detail in section (Blockchain section ..) which is useful in preserving the properties of authenticity and non repudiation.
A cryptosystem can be seen as a five tuple (P,C,K,E,D) that satisfies the following conditions:
P is a finite set of plain texts.
C is a finite set of cipher texts.
K, the keyspace is a finite set of keys
E, set of encryption rules e_k: P $\Rightarrow$ C
D, sete of decrytion rules d_k : C $\Rightarrow$ M.
for each k $\in$ K, there is $e_k \in$ E and $d_k \in$ D such that $d_k(e_k(m))$ = m for every plaintext m $\in$ P.

3.2.2 Hash functions

Cryptographic hash functions are a one-way function, also known as mathematical trapdoor function that transforms an input message into a fixed length binary output. It is one way because although converting a message input to a hash value or a message digest can be done in constant time, reversing the operation is practically impossible to achieve as its computationally inefficient. Earlier hash functions include MD5 which produces a 128-bit hash value but is vulnerable and can be cracked by brute force attack. The predecessors hash functions are sha-256 preceeded by sha-1, sha-2 and others. Their applications include the digital signature, message authentication both of which are interesting for blockchain as will be discussed in section(name). The essential characteristics of hash functions are their deterministic output, meaning given a fixed input; it will always generate the same output. It offers collision resistant property, i.e. it is impossible or extremely rare to get the same hash value for two different messages. If m1 and m2 are the message and h(m1) and h(m2) are hash functions applied to them respectively, collision resistant ensures that h(m1) != h(m2). Another important characteristic of a hash function is that the hash value does not indicate the original information that was hashed thus making it efficient for hiding information.

3.2.3 Digital Signature

A digital signature acts as an intermediary to prove that an entity A, has the password without ever requiring A to reveal it. To create a digital signature, one would need to apply signing algorithm to the private key along with the message. Likewise, anyone can verify the generated signature by applying it to a verification algorithm along with the public key and the message. If a node A intends to send a transaction to B on a blockchain network, A needs to prove that he is the rightful owner of the public address from where the message originated. This is done by creating a digital signature using A's private key from the transaction message. Once the transaction is broadcasted, any node in the network can verify that signature corresponds with A's public key. The signature is dependent on the message, and thus any attempt by a malicious node on the network to modify the message will refute the signature.

3.3 Blockchain Technology

Blockchain Technology is a variant of distributed database on a P2P network. As the name suggests, it is a chain of blocks where each block consists of a list of transactions that are collected by validators/miners defined by the network. The linking and ordering of transactions are also the responsibility of these defined validators. One of the validators proposes a block that the whole network can accept or reject. A consensus algorithm(cite) comes into play during the proposal of a block. If a consensus is reached, then the block is accepted and refused otherwise. Several consensus mechanisms are discussed in section(cite). The proposed block is then chained with the rest of the blocks and linked directly to the block before it. The hash value of the previous block points to the current block and the current block's hash value will point to the future block.
Each block is linked together to a previous block by a hash value(cite figure). Thus, this chain of blocks is unalterable as altering one block would require modifying every other previous block in the chain that is linked together. One could view this order as a singly linked list data structure. The chain of the block shows the ordering of transactions that met consensus at the given time whereas transactions chain shows the chain of ownership. i.e., each transaction that is linked together in a block.

3.3.1 Evolution & Categories

Bitcoin was the first application that made use of Blockchain technology which was a peer-to-peer electronic cash system. The major contribution of this application was solving distributed trust at scale without using a trusted intermediary. Along the dimension of validation and access control(cite), blockchains can be categorized as a public permissionless system, public permissioned, and private permissioned.

- Public Permissionless:Anyone can join the network and become a writer of the block as long as they can solve a problem or reach the consensus that satisfies the underlying protocol. The records are publicly available and thus publicly verifiable.
- Public Permissioned: Anyone can still join the network, but a writer of the block is known but not necessarily trusted entity. The records are publicly verifiable.
- Private Permissioned: This is similar to a Public permissioned setting, but the records are not made public and therefore doesn't offer public verifiability. This

kind of setup is more specific to business use-cases where one business doesn't need to know about other business policies or customer information etc.

[Paper] provides a detailed discussion on various blockchain types and their uses.

3.3.2 Consensus Mechanisms

As a distributed database with multiple writers, there has to be a way for everyone to reach a consensus on a shared global view of the network. Consensus mechanisms allow doing so. Based on consensus mechanisms, systems can be distinctly categorized into [Note: taken from hashgraph presentation slides]

- Leader Based System: In this case, there is a pre-selected leader that collects all the transactions and appends new records to the blockchain. Having a small group or consortium, it has low computational requirements. As a blockchain protocol, it offers an immutable audit of the records. However, just like any other centralized system, this system is susceptible to DDOS attacks and third-party(leader) interference. Generally used in a private or permissioned blockchain setup, it offers higher throughput compared to public permissionless blockchains. Examples include Hyperledger Fabric, R3 Corda, IOTA, etc.

- Proof-Of-Work: This is the most widely used consensus mechanism in a public permissionless setup. As the name suggests, a validator/miner needs to provide the proof to the network that it has done a significant amount of work. This work requires miners to invest a substantial amount of computational resource. The reason for this is that everyone(all miners) compete to be the writer of the next block for which they need to solve a cryptographic puzzle. Mainly, they need to find a hash value that can be associated with the proposed block. The only way to find this value is by brute-forcing. The apparent advantage of such consensus mechanism is that it makes the system DDOS resistant while offering immutable audit trail and scalability. However, miners can still decide upon the order of transactions to include in the block although they cannot modify the transaction. As such, one could term this as 'unfair' since the transaction doesn't get picked up in order of when it was broadcasted to the network.

- Economy based systems: Consensus mechanisms such as Proof-of-stake or delegated proof-of-stake can be seen as an economy based system. Unlike PoW, miners don't compete with each other to be the writer of next block thus saving lots of computational resources. The general idea is that participants can put the respective platform based native token they own at stake to validate a block. Whoever has the highest value at stake gets to write the next block. If the participation turned out to be a malicious one, then all the tokens that were at stake get lost. As such, it puts scarce resource at stake. However, this includes problem such as nothing-at-stake(cite). i.e., a node could vouch for two forks of the same blockchain with nothing to lose. Other drawbacks of this approach are that there is no certainty of consensus, and often has no total ordering of transactions. Examples include Casper, IOTA, etc.

3.3.3 Smart contracts

3.3.4 Applications

4 Methodology and Implementation

The problem of measuring the trustworthiness of communicating entities is an essential aspect of any online system where they interact with each other for any purpose, be it shopping, content delivery or file sharing. This chapter follows on a discussion of a proposed endorsement network where physically or digitally acquainted entities can endorse each other or their presented information. The model will address several concerns such as the roles and requirements of participants as endorser and endorsee, why a participant would play by the rule and what is to stop them from not doing so, threat models, etc. With a system of smart contracts, PoC design will confer interaction between entities, aggregation of information and assignment of scores for final computation. The storage of data both on and off-chain will be discussed.

4.1 Problem Statement

To be able to rely on the trustworthiness of an entity as presented by any online systems, the underlying reputation system needs to be robust and as transparent as possible. The assurance that available information has not been tampered with and correctness of claimed identity should be provided to sustain minimal risk of fraud. The immutable, trustless, decentralized and distributed attribute of blockchain protocol is a recommended solution on a public permissionless network.

4.2 User stories & Requirements

Anyone can join the network and become a participant in the endorsement system. The two notable roles of a user are endorser and endorsee. An endorser can initiate the transaction by sending an endorsement to the participant they wish to. The same user can assume both the roles of endorser and endorsee as long as a set of predefined requirements are met.

The user stories for each role that defines the system requirements for each user type is presented in table 4.2.

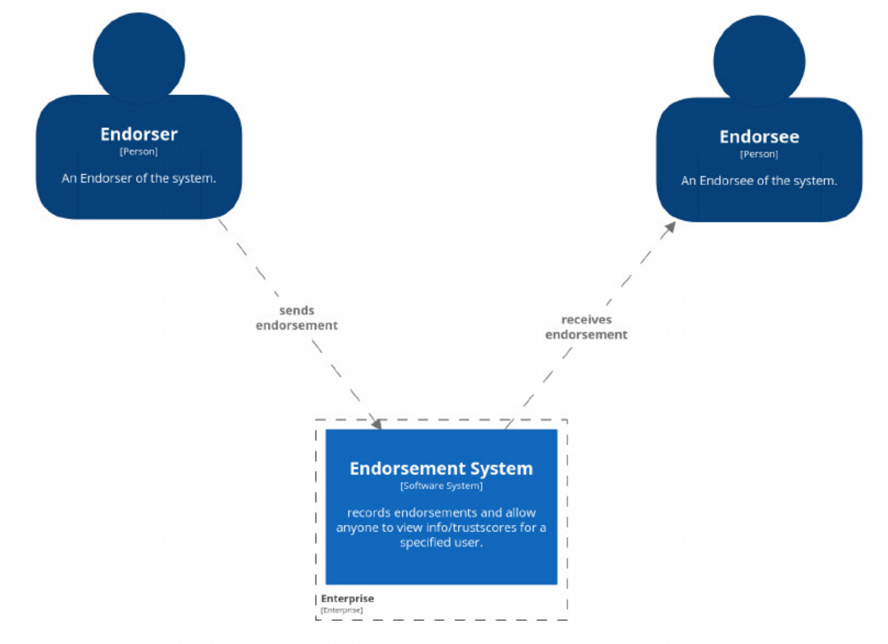

Figure 4.1: Context Layer

As an	I need to be able to..	Traceability
Endorser	send an endorsement so that the endorsement is received by the endorsee.	R1
	remove endorsement so that the endorsement is removed from the endorsee.	R2
	view a list of endorsees so that i can see to whom i have sent endorsements.	R3
	view/edit my personal information so that i can keep it up to date	R5
Endorsee	view a list of endorsers so that I can see from whom I have received endorsements.	R3
other users	compute the total endorsement impact(i.e., final computed score) of any registered members so that I can make an informed decision about the future transactions.	R4
	make a request to join the endorsement network so that I can start sending/receiving endorsements.	R1

The functional requirements can be listed in points as :

1. It must be impossible to make an Endorsement if the endorser and endorsee is same address or not a registered participant.

2. It must be impossible to remove an endorsement if the endorser doesn't belong to the list of endorsers for the given endorsee.
3. All the endorsements must be stored such that, it is possible to see:
   - endorser and endorsee for the given endorsement.
   - degree of incoming and outgoing connections for all endorsers and endorsees.
4. There must be a way to link the public key hashes to the corresponding computed trust scores.
5. It must be possible for a participant to edit their own profile if the editor is the same as the profile owner.

The non-functional system requirements are :

1. Security: smartcontract security.
2. Reliability: reliability of data, tamperproof and verifiable,immutable traceability.

4.3 The Model - Endorsement Network

The initial assumption is that all nodes are honest and as such receive equal points that they can spend at will once registered on the network. This received points are the consumable power that keeps depleting with every endorsement connection made along the way. As depicted in figure 4.2 , these points follow a convergent sequence that converges to the limit 0 as the number of connection 'n' increases. As such, increasing the number of connection alone will not be enough to achieve a higher impact on the network.

4.3.1 Computation of Total Endorsement Impact(tei)

The total endorsement impact corresponds to the total impact a participant has made on the network by sending or receiving endorsements. The two factors that are primarily responsible for computation of final trust score of a participant are the number of incoming and outgoing connections.

The final trust score, associated with the total endorsement impact a participant/node has made on the network requires familiarity with some new terminologies which is briefly discussed below.

nEG_A: number of endorsements sent by a participant 'A'.

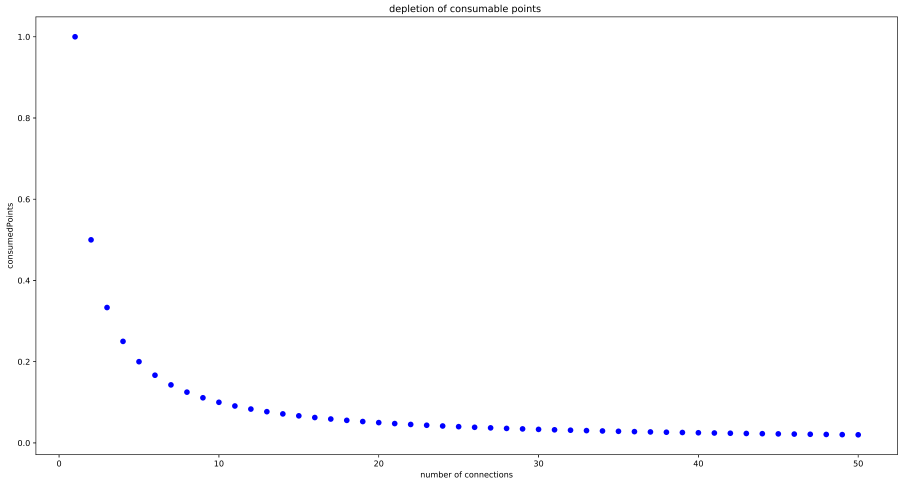

Figure 4.2: Convergent behaviour of consumable points as 'n' increases

nER_A: number of endorsements received by a participant 'A'.

$ratio_A$: ratio of nEG_A to nER_A. This value ensures that sent and received endorsement are not far off from each other. $ratio_A$ is always assumed to be less than 1 and is given by:

$$ratio_A = \frac{min(nEG_A, nER_A)}{max(nEG_A, nER_A)} \tag{4.1}$$

cp_A: Total amounts of points spent by a participant 'A' out of the initially received consumable points. 1 being the initial consumable points received by everyone who joins the network, cp_A is given by $1/nEG_A$.
TRP_A: This corresponds to A's total received points which is the accumulated sum of consumed points by all endorser of A. If a peer 'A' receives an endorsement from 'n' number of peers, then the TRP_A is calculated as:

$$TRP_A = \sum_{i=0}^{n} cp_i \tag{4.2}$$

Finally, the total endorsement impact made by 'A' is given by:

$$TEI_a = ratio_a * cp_a * TRP_a \tag{4.3}$$

4.4 Design of PoC

This chapter focuses on overview and design details of PoC based on the requirements mentioned in section 4.2 on page 13. It starts with design considerations, smart contract

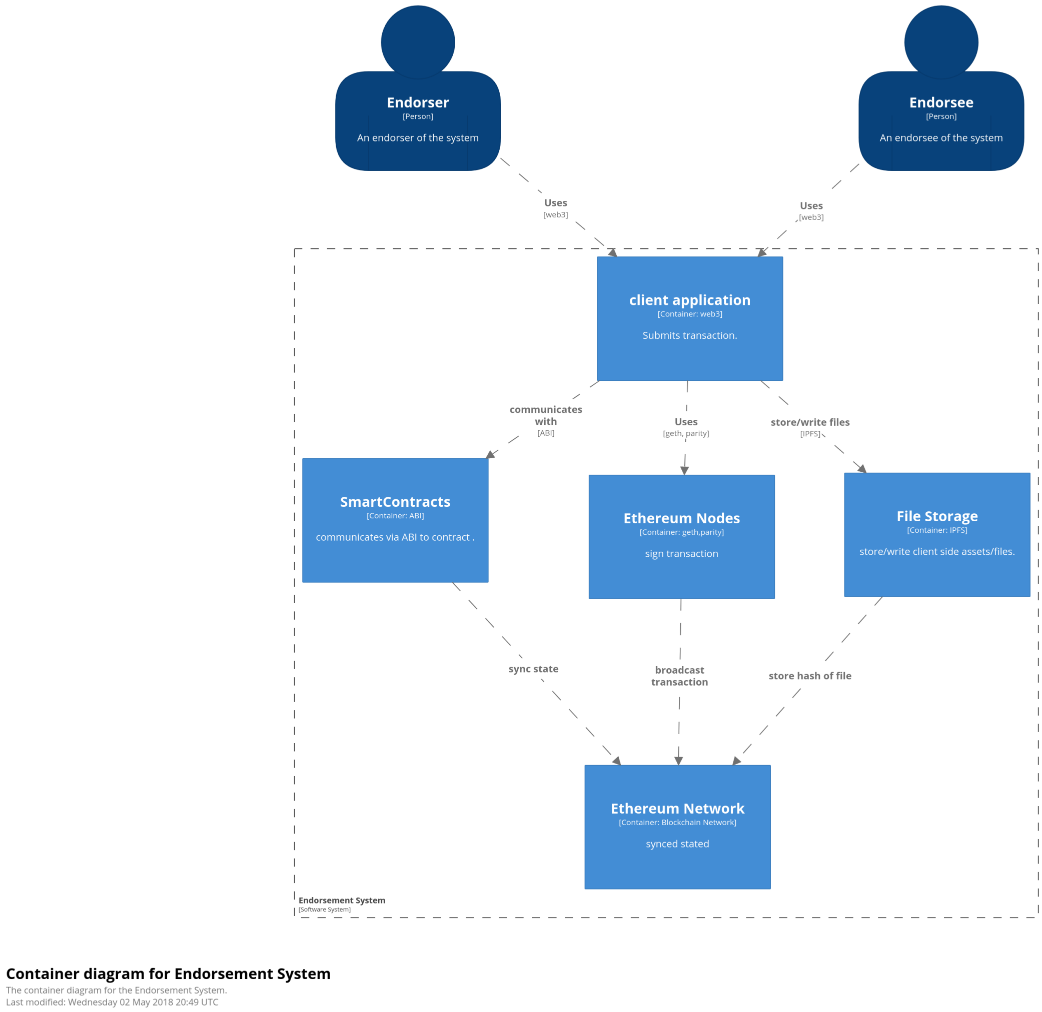

Figure 4.3: Container Layer

setup, data storage on and off blockchain. The high level system overview is presented in 4.3

4.4.1 Design Consideration

Honest and malicious participation in the network and the possible behavior that can result from the interaction between nodes were considered during the design. It is also worth mentioning why a participant would endorse another participant registered on the network. The endorsement is between known entities based on physical and digital acquaintances. The acquaintance could be of the following form:

- Alice and Bob go to the same school/workplace, have worked on multiple projects together and is confident of Bob's reliability.
- Alice has dealt many times with Bob in an online shopping website and always had an excellent transaction with him. In this interaction, Alice is sure that Bob is an honest seller and Bob is confident that Alice is a reliable buyer.

- Alice follows Bob on some social media(e.g.Blog) and knows that Bob's article is good and sees lots of pre-research in his writing and is confident that Bob doesn't engage in the false news.

Thus, Alice is likely to endorse Bob and vice-versa on the endorsement network. They can use secure messaging channels to exchange their keys and endorse each other.

From a game-theoretic perspective of a behavioral outcome, following definition were made for network influencing factors.

1. **Fake endorsements with pseudonymous identities:**Endorsement system being on a distributed, public permissionless blockchain network allows anyone to join and start sending endorsements immediately to whoever they wish to. This creates the possibility that an entity could create multiple pseudonymous identities with an aim to inflate their impact on the network by increasing nEG or nER for their associated persona. There is no straightforward way to detect and stop this behavior right away. However, if doing so doesn't provide any significant advantage, then the assumption is that a rational decision would be not to do it. One factor that is believed to stop a participant from making too many endorsements is the convergent behavior of consumable points. While, there is no limit to the amount of connection a participant can form, as the number of connection increases, the value of consumable points decreases.

2. **Transaction cost:** Ethereum is a programmable blockchain that supports a Turing complete programming language. Thus, to avoid running in infinite loops or DDOS attacks, gas was introduced. Every operation has a gas cost. One could write a program to do anything they wish to do on the network as long as the account that initiates the transaction can pay the gas cost of all operations. The gas consumption is an imperative aspect of endorsement system for two reasons.

    - *Standard transactions on Ethereum:* A participant that makes a call to 'sendEndorsement' function is responsible for paying all the required gas costs. This function updates the state variable nEG and nER. While the price may not seem too high for making one transaction, a malicious node with multiple pseudonymous identities has to pay for all the operations initiated by all personas. For instance, given the interaction graph in the figure, if Alice is an honest node, then she only needs to pay for the operation of one transaction. Whereas if both Bob and Charlie are the pseudonymous identities of Alice, she needs to pay for six transactions. Thus, the assumption is that they may make ether transfer at some point between their accounts. This information is publicly available for anyone on the ethereum blockchain network to view the chain of ownership. If some interactions in the endorsement network look suspicious, one could look up this detail. This method is not guaranteed to detect a Sybil node on the endorsement network but is just another additional factor that might be useful before making a decision.

- *Local information of all neighbouring nodes:* Whenever an endorser makes a new connection, the nEG, and consumable point change accordingly. This change in consumable point has to be reflected for the list of all endorsees. This is not to be confused with a one time update. Every new connection made by an endorser changes the state for all his/her old endorsees. Therefore, all the neighboring nodes of an endorser should be stored previously. The impact of a participant is dependent on his/her direct interaction as well as the endorsers(the participants that have endorsed them). There is no way to make constant cost lookups and updates for this operation. It requires iterating through the list of arrays and computing the impact of every endorsee based on the updated state variable. While it is possible to iterate through items in the array, the general recommendation is to avoid them if possible. One could surely assume that the list will not grow too big for two reasons: (a)A rational node will not make too many connections for reasons mentioned earlier in 1 (b)Dunbar's number suggests a cognitive limit of 150-250 stable social relationships for humans.
  One way to approach this problem is to store the list of endorsees and endorsers for a participant but not change the state. The computation can be done on the client-side using language such as javascript. The final score will be done by the client. However, all the variables necessary to compute the final score will be stored and updated on blockchain as a publicly verifiable information.

3. **Dynamism:** The dynamic social behavior of human is that trust between two entities is not perpetual. Alice may have trusted Bob yesterday but refuses to endorse him today. Trust is dynamic and so is the endorsement decision that an entity can take. Therefore, the design also considers removal of endorsement previously assigned. The removal of endorsement is captured by the figure 4.4.

4. **Free-Riders Problem:** Free riders problem is addressed by making it necessary to maintain the ratio between nEG and nER. A peer without a balanced proportion cannot have a significant impact score on the endorsement network. This method also discourages Sybil nodes because each identity needs to have an almost equal bi-directional connection. If they are only receiving from their own pseudo identity that don't have too many connections, then the impact is ignorant and thus useless and not worth the time.

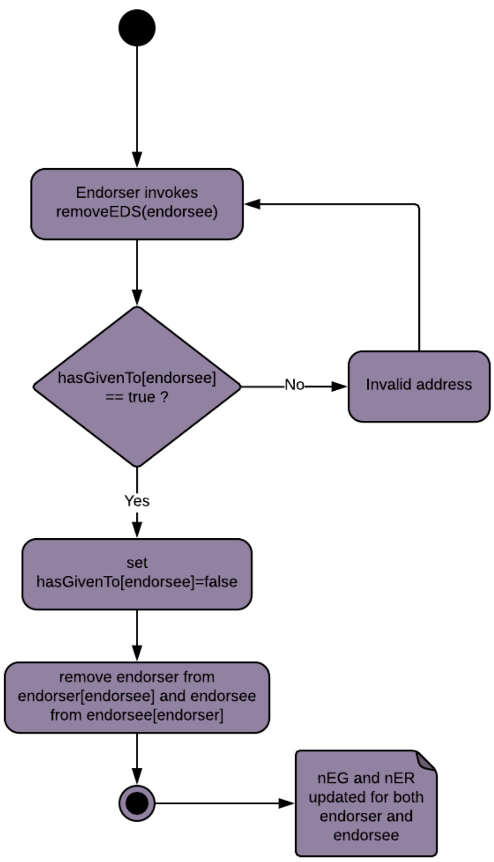

Figure 4.4: Activity diagram for removing endorsement

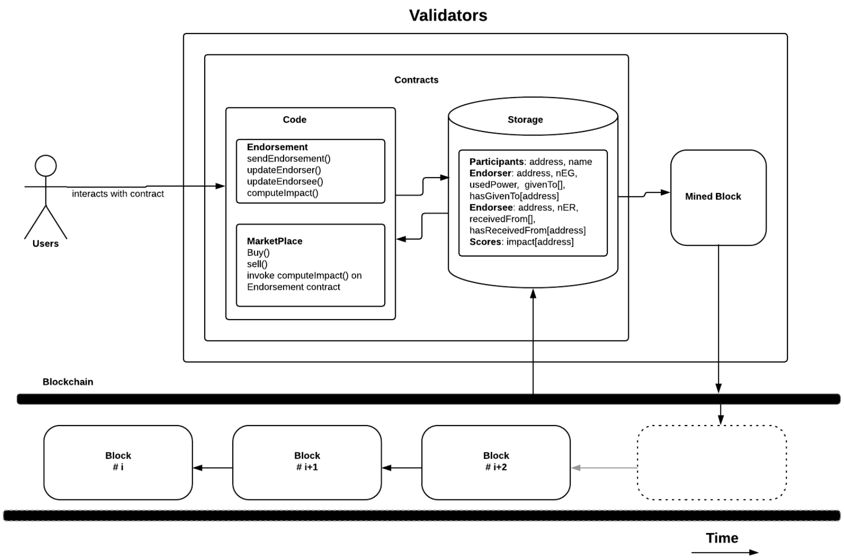

Figure 4.5: Smart contract system

4.4.2 SmartContract

Smart contracts can be either deterministic which doesn't require any information from outside blockchain or non-deterministic which does need to get oracles from external sources.(cite) For this PoC, endorsement contract is deterministic and so all the data and variables required are stored and executed on the blockchain.

The system of smart contracts on the component level is depicted in figure 4.5. The main contracts written for this PoC are :

- Ownable: tracks the owner of the contract. i.e., the creator of contract.
- killable: inherits from ownable and can be killed by owner only.
- Participants: set participant and store their information. An index to access each participant.
- Endorsement: It inherits from participants and can be called by participants only. Endorsement contract handles the core logic of endorsement system, accesses/-queries addresses from Participants and is used for storage of data along with CRUD operations on them.
- Computation: inherits from Endorsement contract and allows anyone to get the final score by accessing the current state from an Endorsement contract. Market-place: stores the buyers, sellers information and allows them to buy or sell the product. Also, allows buyer/seller to compute the score of the involved entity before doing a transaction.

'Marketplace' contract was written to test the endorsement network on a transactional network. However, when deploying endorsement system in the real world, other transactional network/online systems are assumed to have their reputation platform. The reputation platform should have assigned a score to the corresponding users based on the behavior on that network. The endorsement system can act as additional conformity for deciding on a transaction. Say, Alice is registered on Endorsement network and has made a decent score. If she wants to sell a product on Marketplace(or any other transactional network), she can claim about her score and anyone who wants to buy from Alice can verify the claim by checking the score that corresponds to her public address. If both Alice and buyer are registered on the endorsement network on the blockchain, they can send a pre-transaction message to each other to verify that Alice is who she claims to be and vice-versa. In case the buyer is not registered on the endorsement network then Alice can prove the claim by signing a cryptographic challenge with her private key.

4.4.3 Data and variables on and off blockchain

For this PoC, the data required to identify the users are stored on the blockchain. But, it does preserve the anonymity requirement mentioned in section 4.2, as the only public information is the link between the public key hash and individual trust score. Even though the users are required to register with a pseudonym, it is not needed that the aliases be linked to real-world identity. A user might like to share more information(other online account ids, address, etc.). As mentioned earlier in the gas consumption section, every non-zero byte data or code of a transaction costs a certain amount of gas. (cite: eth yellow paper)(cite: eth gas station chart on the gas section above). Storing this data can become an expensive operation for real-world usage. The right approach can be to use an off-blockchain storage solution such as IPFS, swarm (cite). The hash that points to the file in IPFS can then be stored on the blockchain. Generally, client-side assets (HTML, js) are stored on these distributed off-chain file system that can communicate to the contracts registered on blockchain network.

4.4.4 Blockchain and Consensus algorithms

The proposed platform for this PoC is Ethereum, constantly developing open source blockchain ecosystem. The process of starting up the nodes and deploying the contract on the network is shown in figure 4.6. As a permissionless system, it allows any nodes to collect transactions and act as a writer. Consensus mechanism that is generally used in a permissionless setting is Proof-of-Work or Proof-of-Stake. As mentioned earlier, PoW is computationally expensive and wasteful. Using other consensus mechanisms such as delegated proof of stake requires finding enough trustworthy validators that can act as a leader or a master node which can be given authority to vote on behalf of the community. In the endorsement network, an honest endorsement is supposed to be between known entities. But all participating entities do not know each other and there is no one node

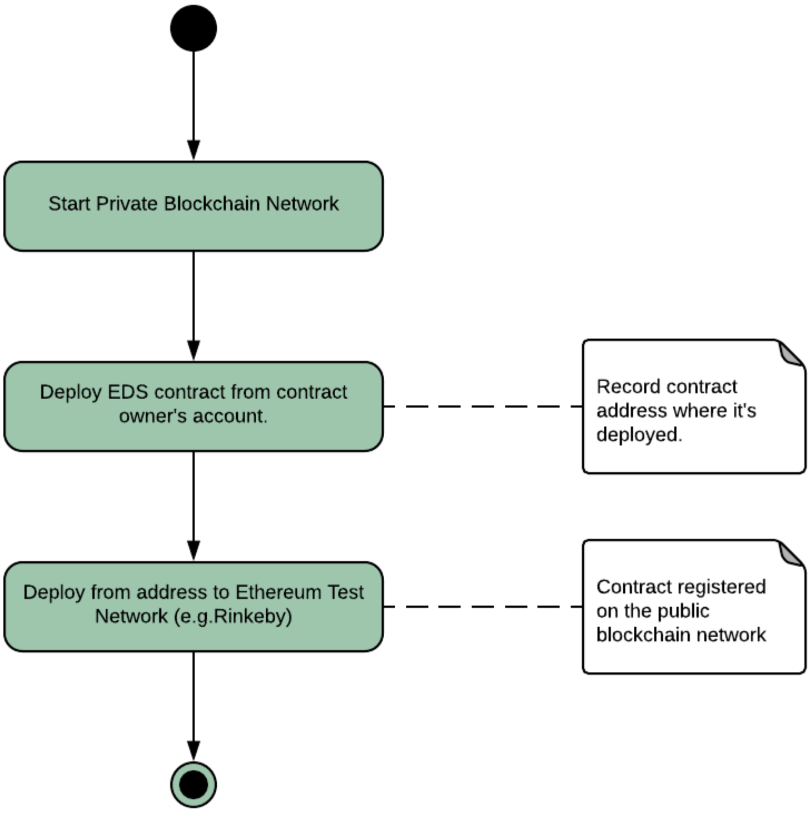

Figure 4.6: Startup activity for registering contract on the network

that can be trusted to collect everyone's transaction and make final commit. It is more like a cluster of small-world graph topology(cite graph section). The recent breakthrough in consensus engine is proposed by hashgraph(cite) that claims to be fair(no bias as in other systems where miners get to decide which transactions to include in the block) and provide 100% certainty on the order of transactions with instant validation(cite). The hashgraph conundrum is that their software is patented and a developer must pay for making an API call using micropayment of the platform. The suggested approach for this PoC would be to use a public blockchain but in a permissioned setting much like Sovrin(cite) does. Using known but not necessarily trusted writers and still allowing public verifiability. This is more sensible for a P2P network that provides services to end users or any centralized/decentralized transactional network such as e-commerce, filesharing platform, etc.

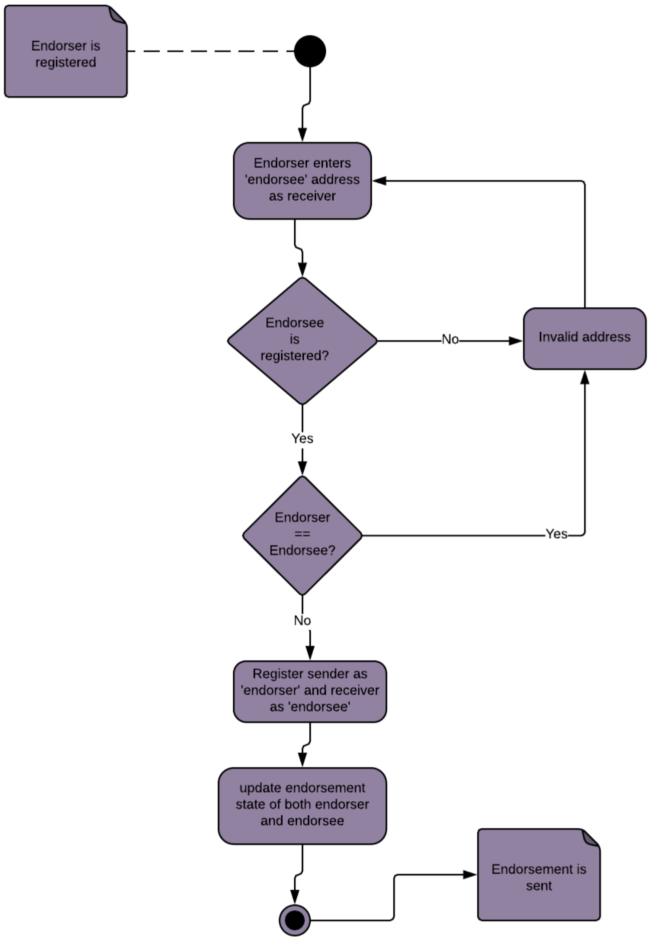

Figure 4.7: Activity Diagram for sending an endorsement

5 Results

5.1 Interaction graph

A participant sends an endorsement to their acquaintances. There is no absolute way to find out if it's an honest interaction or not. i.e., if the participating nodes are the distinct or pseudonymous identity of the same node. The only information visible about the participants on the network are the public address and information they chose to disclose to selected members. Therefore, an ideal way is to view the endorsements as an interaction graph where nodes are entities participating and edges define the interaction along with the direction. Graph algorithms can help to determine the anamoly in the network. Using Net flow rate convergence, anamoly detection is simplified and explained further.

5.2 Analysis

5.3 Measurement

5.4 Comparison

6 Discussion & Analysis

6.1 Generalization

6.2 first section

The results presented in Chapter 4 are discussed and analyzed, including comments and reflections from the author. It may include the following: Comparison of obtained results with discussion,interpretation and evaluation of results. Results of analysis or modeling are described. Interpretations are drawn and connected to previous work

7 Conclusion

7.1 first section

7.1.1 first subsection

Synopsis of findings, limitations, further proposals for future work on the subject. Clear conclusions are drawn that stem from the previous analysis. Present the conclusions drawn and the evidence and arguments that support the conclusions.
Do not include new findings, but only refer to results already discussed in the thesis. Relevant further work in the field is summarized.

Literature

[1] D. H. McKnight and N. L. Chervany, "The meanings of trust", 1996.

[2] ——, "Trust and distrust definitions: One bite at a time", in *Trust in Cyber-societies*, Springer, 2001, pp. 27–54.

[3] D. Gambetta *et al.*, "Can we trust trust", *Trust: Making and breaking cooperative relations*, vol. 13, pp. 213–237, 2000.

[4] G. Zacharia, A. Moukas, and P. Maes, "Collaborative reputation mechanisms for electronic marketplaces", *Decision support systems*, vol. 29, no. 4, pp. 371–388, 2000.

[5] J. Sabater and C. Sierra, "Review on computational trust and reputation models", *Artificial Intelligence Review*, vol. 24, no. 1, pp. 33–60, 2005.

[6] C. Castelfranchi and R. Falcone, "Trust and control: A dialectic link", *Applied Artificial Intelligence*, vol. 14, no. 8, pp. 799–823, 2000.

[7] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision", *Decision support systems*, vol. 43, no. 2, pp. 618–644, 2007.

[8] L. Rasmusson and S. Jansson, "Simulated social control for secure internet commerce", in *Proceedings of the 1996 workshop on New security paradigms*, ACM, 1996, pp. 18–25.

[9] L. Mui, M. Mohtashemi, and A. Halberstadt, "Notions of reputation in multi-agents systems: A review", in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, ACM, 2002, pp. 280–287.

[10] M. Carbone, M. Nielsen, and V. Sassone, "A formal model for trust in dynamic networks", in *Software Engineering and Formal Methods, 2003. Proceedings. First International Conference on*, IEEE, 2003, pp. 54–61.

[11] N. Satoshi, "Bitcoin: A peer-to-peer electronic cash system", Accessed 23 Dec 2017, [Online]. Available: https://bitcoin.org/bitcoin.pdf, Bitcoin, Oct 31, 2008.

[12] K. W. Miller, J. Voas, and P. Laplante, "In trust we trust", *Computer*, vol. 43, no. 10, pp. 85–87, 2010.

[13] M. Swan, *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015.

[14] P. Resnick and R. Zeckhauser, "Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system", in *The Economics of the Internet and E-commerce*, Emerald Group Publishing Limited, 2002, pp. 127–157.

[15] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood, "The value of reputation on ebay: A controlled experiment", *Experimental economics*, vol. 9, no. 2, pp. 79–101, 2006.

[16] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems", *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[17] D. B. DeFigueiredo and E. T. Barr, "Trustdavis: A non-exploitable online reputation system", in *E-Commerce Technology, 2005. CEC 2005. Seventh IEEE International Conference on*, IEEE, 2005, pp. 274–283.

[18] A. Schaub, R. Bazin, O. Hasan, and L. Brunie, "A trustless privacy-preserving reputation system", in *IFIP International Information Security and Privacy Conference*, Springer, 2016, pp. 398–411.

[19] J. A. Bondy, U. S. R. Murty, *et al.*, *Graph theory with applications*. Citeseer, 1976, vol. 290.

[20] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks", in *Proceedings of the 12th international conference on World Wide Web*, ACM, 2003, pp. 640–651.