

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

June 4, 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

June 4, 2018

Abstract

Abstract here

Acknowledgements

Acknowledgements here

Table of Contents

Abstract	II
Acknowledgements	III
List of Tables	VI
List of Figures	VII
List of Abbreviations	VIII
1 Introduction	1
1.1 Definition	1
1.1.1 Trust and Reputation	1
1.1.2 Blockchain	2
1.2 Motivation	3
1.3 Purpose and research questions	4
1.4 Scope	4
1.5 Structure of Report	5
1.5.1 acronyms	5
2 Literature Review	6
2.1 Reputation Algorithms	7
3 Background	8
3.1 Graph properties	8
3.2 Cryptography	9
3.2.1 Basic Concepts	9
3.2.2 Hash functions	9
3.2.3 Digital Signature	10
3.3 Blockchain Technology	11
3.3.1 Evolution & Categories	11
3.3.2 Consensus Mechanisms	12
3.3.3 Smart contracts	13
3.3.4 Applications	14
4 Methodology and Implementation	15
4.1 Problem Statement	15

4.2	User stories & Requirements	15
4.3	The Model - Endorsement Network	17
4.3.1	Computation of Total Endorsement Impact(TEI)	17
4.4	Design of PoC	18
4.4.1	Design Consideration	19
4.4.2	SmartContract	22
4.4.3	Data and variables on and off blockchain	24
4.4.4	Blockchain and Consensus algorithms	24
5	Results	28
5.1	Interaction graph	28
5.2	Impact and Ratio: Case-by-case across different scenarios	31
6	Discussion & Analysis	35
6.1	Generalization	35
6.2	first section	35
7	Conclusion	36
7.1	first section	36
7.1.1	first subsection	36
	Literature	37

List of Tables

Table 5.1: Nodes with Impact zero because of the received points	29
Table 5.2: No. of nodes and the corresponding impact ranges	31

List of Figures

Figure 4.1: Context Layer	16
Figure 4.2: Convergent behaviour of consumable points as 'n' increases	18
Figure 4.3: Container Layer	19
Figure 4.4: Activity diagram for removing endorsement	23
Figure 4.5: Smart contract system	24
Figure 4.6: Startup activity for registering contract on the network	25
Figure 4.7: Activity Diagram for sending an endorsement	27
Figure 5.1: Given Vs. Received	28
Figure 5.2: Received Points Vs. Total Endorsement Impact	29
Figure 5.3: Interaction subgraph of nodes with impact zero	30
Figure 5.4: Relation of Ratio and Total endorsement impact	31
Figure 5.5: Case1- Maximum Impact Minimum Ratio	32
Figure 5.6: Case2- Maximum Impact Maximum Ratio	33
Figure 5.7: Case3- Minimum Impact Minimum Ratio	33
Figure 5.8: Case4- Minimum Impact Maximum Ratio	34

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 rules¹. 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 trust-less 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 such as the ones mentioned in section 3.2 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 Experian⁵ and Javelin⁶ 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 exposed⁹ 130,000 users login details while Facebook has admitted to the compromise of 2.2 billion of its user's data¹⁰. 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

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-time-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>

algorithms with Blockchain technology could help to ensure trustworthiness of online entities with correctness of data and a high degree of accuracy.

1.3 Purpose and research questions

The primary 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 will be quantified to infer reputation score which in turn can yield a value that can represent the impact the agent has made on the network. The nodes and their relationship will be studied to analyze honest or malicious participation. Generalization of this endorsement network to serve other use cases shall be discussed as well.

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.

Research Question 1

To answer research question 1, literature survey will be performed on various reputation algorithms and existing models. This survey should follow with the discussion on various analysis metrics and threat models eventually leading to graph simulation of endorsement network.

Research Question 2

Interpretation of nodes connections and quantification of scores for individual nodes that represents trustworthiness based on score range will be presented. Comparative analysis of on chain vs. off-chain storage requirements will be studied and analyzed. Overall system design and architecture will be presented.

Research Question 3

The endorsement network will be analyzed against various network metrics to show

resilience to threat models. Discussion on other use cases and how the endorsement model can be used on top of other systems will be presented.

1.5 Structure of Report

This paper is structured as follows. Chapter 2 will perform a literature survey on the existing algorithms and their implementations. Chapter 3 will provide a background overview of relevant concepts necessary to understand the following sections. In chapter `refch:method`, system requirements and the approach taken for the model design is shown. It will show the overall system design and architecture. Chapter 5 will discuss the test methods and present results representative of the designed model. Finally, conclusion and future works shall be presented in chapter 7.

1.5.1 acronyms

2 Literature Review

The earliest and most known internet reputation is that of eBay¹. 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), whitewashing etc.. There are many none E-commerce online systems such as StackExchange², Yelp³, Quora⁴, Reddit⁵ 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. A single point of control is a single point of failure. In light of this, there have been several research and proposals on decentralized reputation methods for the distributed network(cite). Especially significant for P2P systems such as file-sharing, content delivery applications, etc for detecting the quality of file/content and the owner of those files.

Resnick et al.[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. TrustDavis [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 recommends the use of blind signature(cite) 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

¹<https://www.ebay.com/>

²<https://stackexchange.com/>

³<https://www.yelp.com/>

⁴<https://www.quora.com/>

⁵<https://www.reddit.com/>

service provider's reputation despite their honest behavior in the network. Another blockchain based approach [19]

2.1 Reputation Algorithms

The most known and widely used reputation algorithm in a P2P network is EigenTrust[20] which recommends a method to aggregate local trust values of all peers. It uses the notion of transitive trust. i.e., If a peer 'i' trusts a peer 'j' and peer 'j' trusts peer 'k' then 'i' would also trust 'k'. Peers can rate another peer as either positive or negative [-1, +1]. The users can decide if a peer can be considered trustworthy as a download source based on its total aggregated score. EigenTrust defines five issues that any P2P reputation system should consider. They are self-policing(i.e., enforced by peers and not some central authority), anonymity(i.e., peer's reputation should only be linked to an opaque identifier), no profit to newcomers, minimal overhead for computation/storage and robust to malicious collectives of peers. A significant disadvantage of this algorithm is that peers are more likely to center around a static set of pre-trusted peers that joined the network and thus has limited reliability if they leave the network. Pre-trusted peer is a notion of trust where few peers that join the network are assumed trustworthy by design. Advogato's trust metric[21] also uses this notion. HonestPeer[22] proposes to enhance EigenTrust algorithm further by selecting reputable nodes dynamically and not just relying on pre-trusted nodes. It claims to have a better success rate in quality file serving and lower malicious participation compared to EigenTrust. As mentioned in [23], a reputation system can serve one of the following purpose, a peer-based reputation system or file-based reputation system. The peer-based system allows peers in the network to be rated and assigned a value. A file-based system is concerned more with the integrity of a file that is being delivered/served on the network regardless of who(peer) owns or serves it. AuthenticPeer++[23] is a trust management system for P2P networks that combines both. i.e., it shares the notion of both trusted peers and trusted files. As such, it only allows trusted peers to rank the file after they have downloaded it and uses a DHT-based structure to manage the integrity of file information. Bartercast[24] is a distributed reputation mechanism designed for P2P file-sharing systems. It creates a graph based on data transfers between peers and uses max flow algorithm to compute the reputation values for each node. Tribler⁶ is a BitTorrent based torrent client that uses Bartercast to rank its peers. PowerTrust[25] proposes a robust and scalable reputation system that makes use of Bayesian learning and lookahead random walk(cite). It uses Bayesian method to generate local trust scores and a distributed ranking mechanism to aggregate reputation scores.

⁶<https://www.tribler.org/>

3 Background

This chapter provides an overview of the basic concepts necessary to understand the subsequent sections.

3.1 Graph properties

A graph, as the name suggests can be used to represent objects and their relationships graphically. Formally, 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 .

$\varphi_G(e) = uv$ represents that e is an edge that joins vertices u and v [26]. Based on these properties, any online interaction system can be modeled graphically including reputation system. Each node on the network can represent agents/users that interact with other users. This interaction can represent the relationship between nodes as the edges connecting vertices. The transfer of data between the nodes can be quantified to represent the weight of the edge. This weight value can be used to determine the strength or weakness of relationship between the nodes. Modeling the interaction as a graph can help to understand and analyze its complexity at micro, meso, and macro level[27]. At a micro-level, the local properties of a particular node such as its activity, connection degree, its neighbors, and interactions can be observed. On a meso-level, the node's relative position in a given graph that can help to determine its centrality and connectivity can be observed. On a macro-level, one can view the overall network as the topology of the graph itself. It helps to study the global properties of the graph.

Network metrics that are helpful in analyzing the complexity of interactions at macro and meso-level are(cite):

Degree Connectivity: The number of connections a node has is the degree of its connectivity. The number of inflow is referred to as indegree whereas the number of outflows is the outdegree of a node. Usually, a higher degree of connectivity implies a higher likelihood for information(relevant data to the network) to pass through that node.

Network Centrality: Centrality refers to the significance of a node in the network. i.e., how important the node is in the overall network. The degree of connectivity is one way to measure centrality of a node. Similarly, there are other centrality measures which include: **Closeness centrality**, **Betweenness centrality**, **Prestige centrality**.

Closeness centrality refers to how close a node is to other nodes in the network. Betweenness centrality refers to the number of nodes to which the given node acts as a

connector. i.e., how many nodes passes through this node. Prestige centrality refers to the significance of the node based on the significance of the adjacent nodes(nodes one is connected to). To observe and analyze the behavior at the macro-level, one needs to look at the overall structure of the graph. i.e., Network topology that shows how constituent parts are interconnected to form the graph as a whole. They can form ring, star, tree, or mesh structure or be a fully connected graph where each node are connected to each other.

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/encrypted such that only the authorized parties can make sense out of it. i.e., decrypt using the previously agreed upon key. Authenticity relates to confirming the truth of an attribute claimed by an entity. Non-repudiation is associated with the property that any entity who has previously sent the message cannot deny their authorship[28].

A cryptosystem can be defined as a five-tuple (P, C, K, E, D) where,

P , is a finite set of plaintexts and C , a finite set of ciphertexts.

E , set of encryption rules such that $e_k : P \Rightarrow C$.

D , set of decryption rules such that $d_k : C \Rightarrow P$

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 $m \in P$.

A cryptosystem can be either symmetric or asymmetric. Symmetric makes use of the same key for both encryption and decryption whereas asymmetric, also known as public-key cryptography makes use of key pairs, public and private keys. The public key can be publicly distributed, and an entity 'A' wishing to send a confidential message to other entity 'B' can encrypt the message using B's public key k_p to form ciphertext c . Upon receiving c , B can decrypt it using the private key k_s that is only known to B and corresponds to the respective public key that was used to form c . The computation of k_s given k_p is computationally infeasible in a secure system. Besides public-key encryption, public-key cryptography also has use in digital signature as discussed in section 3.2.3. RSA[29] is one example of a public-private cryptosystem.

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. An important characteristics of hash function is its deterministic output. i.e., given an input, it will always produce the same output. This attribute contributes to data verifiability as anyone can always verify if the produced hash output for data matches by simply applying the data to the respective hash function. Other significant properties of hash function that contributes to reliability in digital security are[30]:

One-way: Given a key k , and an output w , it should be hard for an attacker to find x such that the hash of x applied with k , produces w . ie., $H_k(x) = w$.

second pre-image resistant: Given a key k and a string x , it should be hard for an attacker to find y such that $H_k(x) = H_k(y)$.

Collision-resistant: Given a key k , it should be hard for an attacker to find x and y such that $H_k(x) = H_k(y)$.

Earlier hash functions include MD5 which produced an output of size 128 bits. Collision-resistant attack in MD5 is possible within seconds with a complexity of 2^{24} [31]. NIST published secure hashing algorithm (SHA) in 1993 as the secure hash standard. Currently, SHA-3 is the latest in SHA family of standards that was released in 2015 with SHA-0, SHA-1 and SHA-2 as it's predecessor algorithms. Collision attack for SHA-1 was shown to be practically possible[32] by creating two colliding pdf files that produced same SHA-1 digest. It took equivalent processing power of 6,500 years of single CPU computation time and 110 years of GPU computation time. Applications of hashing algorithms include digital signature, ISO checksums, fingerprinting data etc. This cryptographic hash functions is relevant for understanding blockchain data structure. Especially significant for verification of transactions, creation of blocks, linking them to the existing chain. The consensus mechanism, namely PoW relies on SHA-256 hash(in Bitcoin) applied to the block header and nonce to produce a verifiable fingerprint of data. This is discussed in more detail in 3.3. Proof-of-work was originally used in 1997 by Adam Back as a anti-spam system for the then proposed Hashcash[33]. The general idea being that the sender of message would have to compute some number of sha operations before sending the message. The generated fingerprint could be checked by anyone to see if the sender has actually done the required number of computations. If a legitimate user sending out one email had to spend 'x' seconds to do the sha operations, a malicious user intending to send thousands of emails had to spend $1000x$ seconds.

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. As discussed earlier, public-key cryptography uses key-pairs that correspond with each other. In context of Blockchain, If Alice wants to send a value(transaction) to Bob, she can create a transaction message, sign it using her private key and broadcast the transaction over the network. Her signature and the transaction message will be publicly available on the network(assuming a public

blockchain network). Anyone on the network can verify that the signature corresponds to Alice's public key. Thus, Alice can always prove that she is the owner of the public key from where the message originated. The signature is dependent on the message, and therefore any attempt to modify the message will refute the signature.

3.3 Blockchain Technology

Blockchain Technology is a variant of distributed database implemented on a P2P network. Every participating node in the network has the same copy of history of records of database. Blockchain is essentially a chain of blocks, as the name suggests. Each block consists of list of valid transactions (signed messages) collected by validators/miners of the network. The linking and ordering of transactions are also the responsibility of validators. The validators propose a block they have unlocked to the network which can either be accepted or rejected. If accepted, the newly created block gets linked to the existing blockchain with a hash pointer pointing to the previous block. As such, blockchain provides the transaction ordering that every node agreed on at a given time. Consensus metric helps to establish and maintain the integrity of a blockchain system (cite). The primary attributes that constitute a blockchain system are distributed, decentralized, time-stamped transactions. There have been P2P protocols deployed as a file-sharing or other form of content delivery services before Blockchain. What separates Blockchain from them is that for the first time, it made possible to transfer values online on a P2P network without a double-spending problem. i.e., If a peer 'A' sends a file or a value 'v' to 'B,' then the file should be owned by 'B' and removed from the account of A. 'A' should not be able to send the same file 'v' to other entities on the network. This was the significant contribution of the technology.

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 a 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 into¹:

- **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 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

¹Hashgraph

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 etc.

- Hashgraph

3.3.3 Smart contracts

A contract in a classical sense is a set of rules with pre-defined obligations and permissions that participants are assumed to follow. A smart contract, however is a computer program that can codify the interaction between participating entities and self-executes when triggered by an event. It doesn't necessarily need to be legally binding or even associated with the outside world.² The term Smart contract was first coined by Cryptographer Nick Szabo, in 1994 [34] and defined as a computerized transaction protocol that can execute the terms of a contract. Szabo points out that the contract design should fulfill four objectives[35]:

Observability, ability to observe the performance of principal (agents who have agreed to the contract) and prove their performance.

Verifiability, the ability of principals to prove to the arbitrators that the contract has been performed or breached.

Privity, to ensure that the third party should not have control or knowledge of the content or performance. It correlates to both privacy and confidentiality of principals of contract and the contract itself.

Enforceability, to make the contract self-enforcing which can be attributed to by verifiability, built-in incentives mechanism, and objective mentioned above.

Privity refers to minimization of third-party vulnerability by limiting knowledge and control whereas, on the other end, observability and verifiability demand invoking it to an extent. As such, a trade-off is required wherein an optimal balance between these objectives should meet. Thus, trusted intermediaries were introduced with minimal control/observability. However, privity was not guaranteed in case of dispute. [36]. Following the invention of Bitcoin and several decentralized blockchains, the definition of smart contracts have evolved. Ethereum being the first platform to offer programmable blockchains, introduced a virtual machine, EVM, where the contract code can be executed that results in a deterministic output provided the same transaction context and blockchain state.³ EVM often referred to as a single world computer runs on every ethereum node and given the same initial state produces the same final state. Several high level languages can be used to write smartcontracts for different blockchain platform. Examples include solidity, serpent, LLL, etc. For the sake of relevance to this project, solidity as the smart contract language and Ethereum as the blockchain will be used as a point of view henceforth. Contract's code resides in the blockchain as an

²<https://universe.ida.dk/media/23422289/fritz-henglein.pdf>

³<https://github.com/ethereumbook/ethereumbook/blob/develop/smart-contracts.asciidoc>

immutable form. They are not autonomous self-executing program but rather needs to be called by a transaction or invoked by other contracts. Once the code is registered and deployed on the blockchain, its code cannot be altered by anyone, including the owner of the contract. However, there exists a possibility to include killable function that can be executed by the owner which when called executes an EVM opcode called SELF-DESTRUCT and deletes the contract from the blockchain. As in any Turing-complete language, solidity is affected by the halting problem. To address this, Ethereum introduces the concept of gas. To store any state, or execute any operation, gas needs to be supplied. Thus, a program that has a bug or a non-terminating intention will eventually run out of gas and stop.[37]

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.

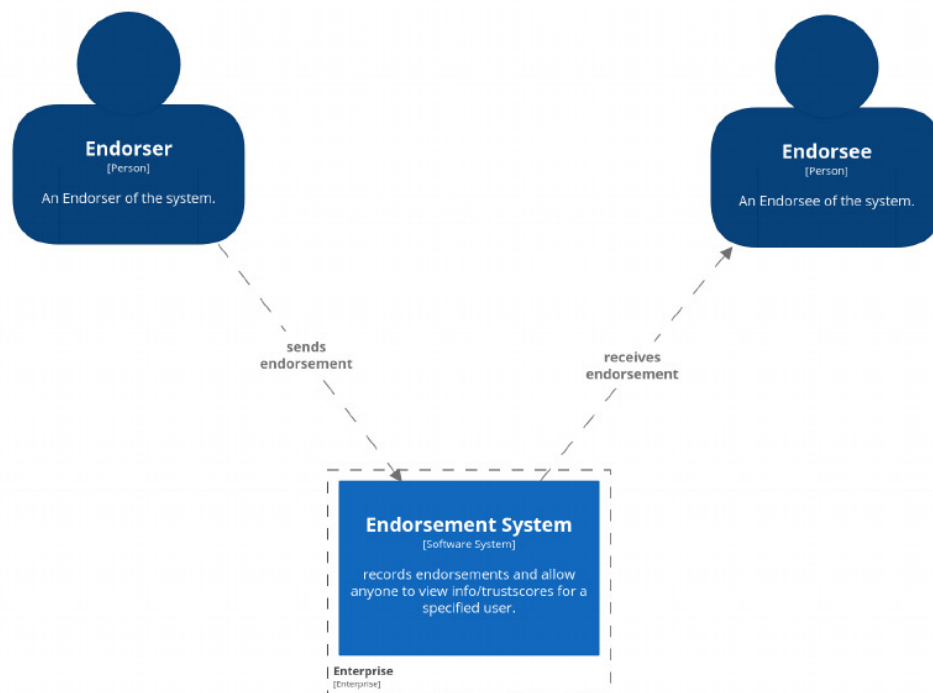


Figure 4.1: Context Layer

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 endorseees.
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.
3. TrustMetrics: TEI should be representative of the trust metrics.

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'.

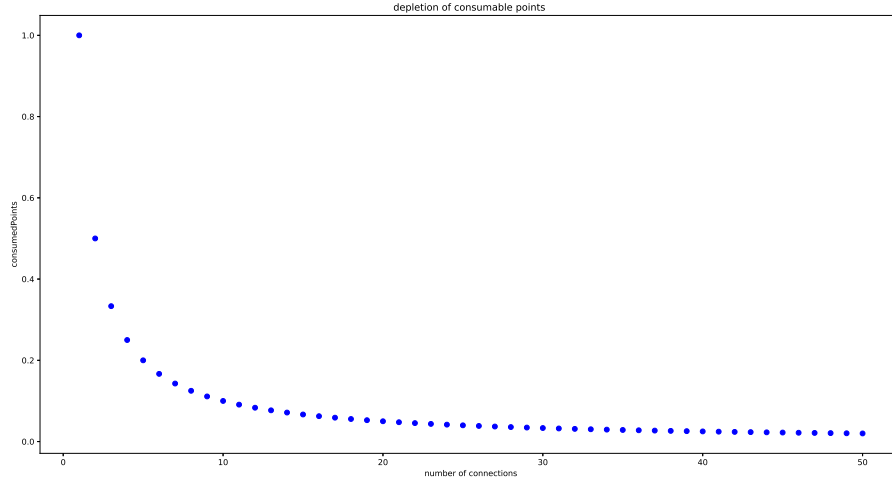


Figure 4.2: Convergent behaviour of consumable points as 'n' increases

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:

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

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:

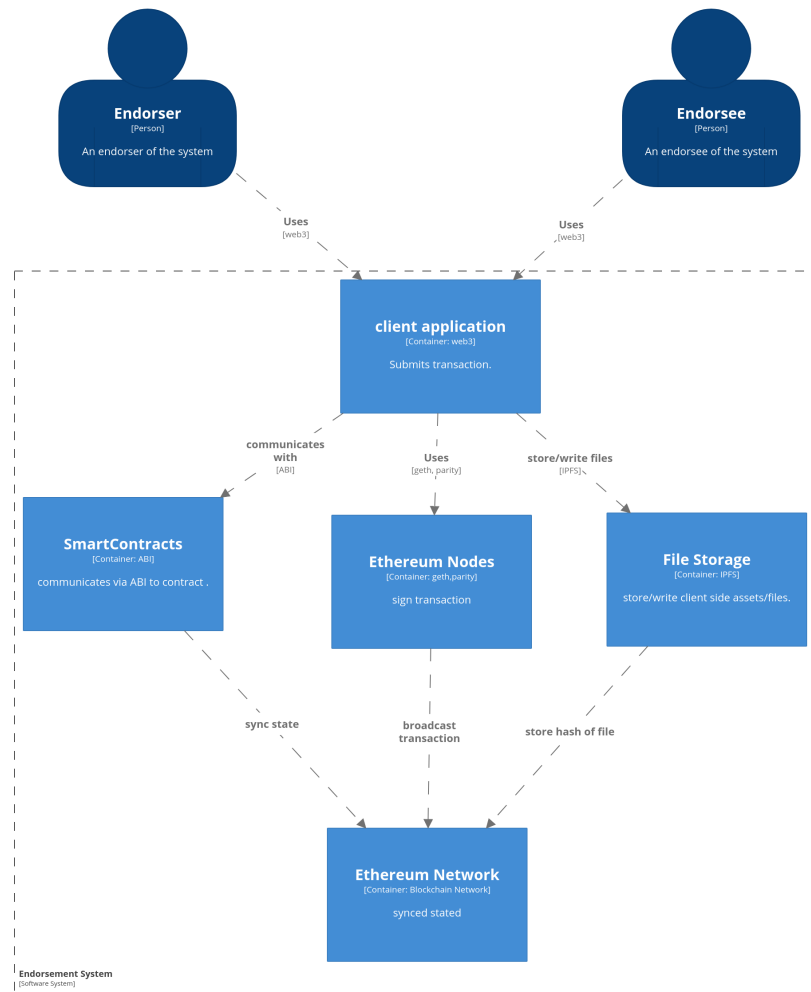
$$TRP_A = \sum_{i=0}^n cp_i \quad (4.2)$$

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

$$TEI_a = ratio_a * cp_a * TRP_a \quad (4.3)$$

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 15. It starts with design considerations, smart contract



Container diagram for Endorsement System

The container diagram for the Endorsement System.
Last modified: Wednesday 02 May 2018 20:49 UTC

Figure 4.3: Container Layer

setup, data storage on and off blockchain. The high level system overview is presented in figure 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 some form of 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 endorser(s) (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 endorser(s) 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.

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. Marketplace: 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.

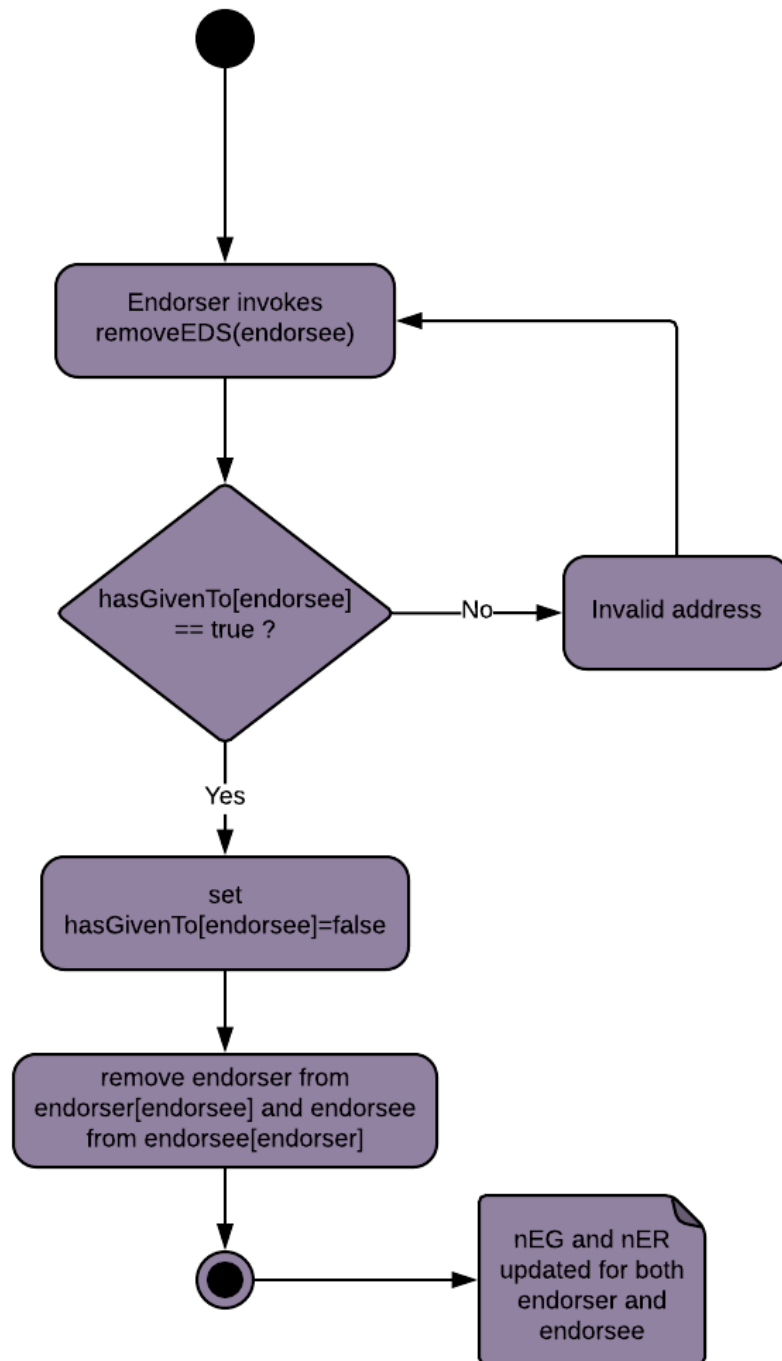


Figure 4.4: Activity diagram for removing endorsement



Figure 4.5: Smart contract system

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

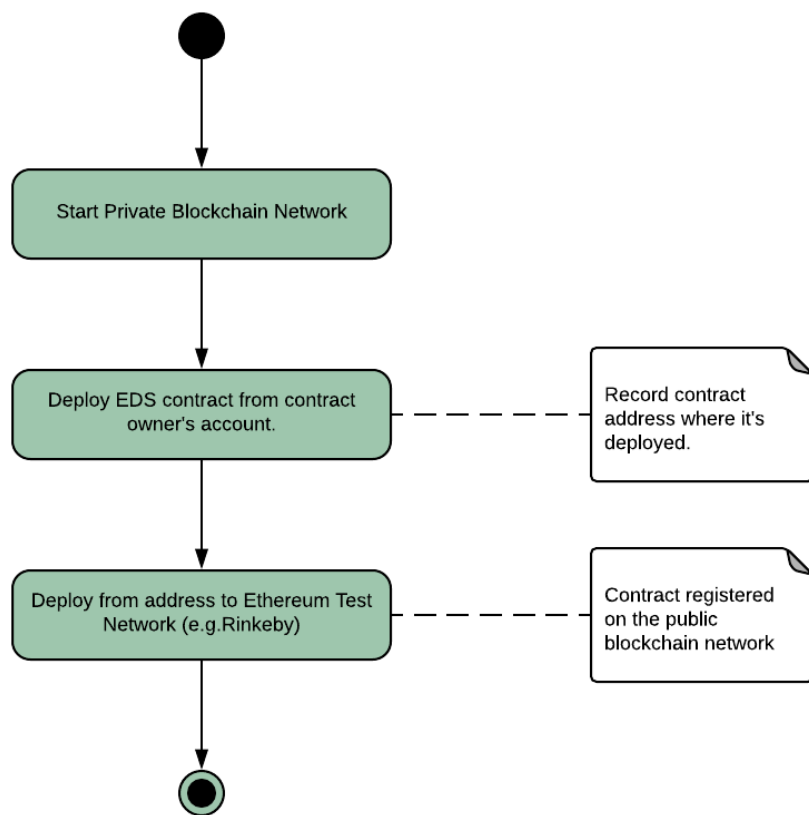


Figure 4.6: Startup activity for registering contract on the network

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 that can be trusted to collect everyone's transaction and make final commit. 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.

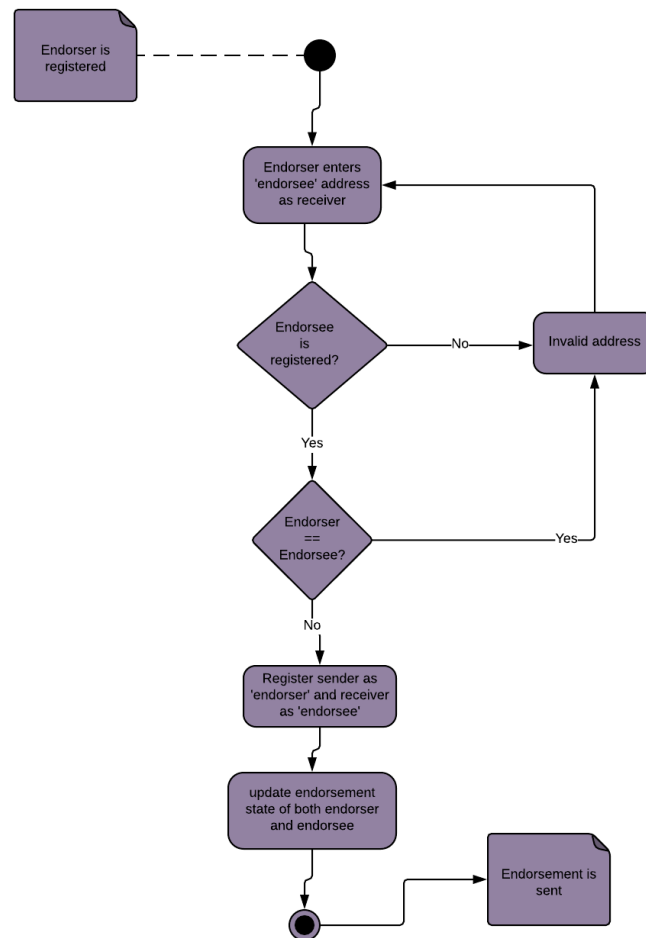


Figure 4.7: Activity Diagram for sending an endorsement

5 Results

This chapter evaluates the result of endorsement model as presented in section 4 by using interaction graph on existing dataset. Results will be presented along with the discussion of measurement metrics and analyzed to see if the requirements mentioned in section 4.2 are met.

5.1 Interaction graph

In order to simulate the interaction graph, existing dataset from SNAP [38] was used. The dataset was extracted from Bitcoin Alpha trust ¹ weighted signed network which was essentially a who-trusts-whom network of people that trade on Bitcoin Alpha platform. Participants on this network rated each other on a scale of -10 to +10 where negative value represented total distrust whereas positive value represented total trust. It consists of 3,783 nodes that made 24,186 edges out of which 93% of the edges were marked as positive edges[39].

The available information in the dataset for all the nodes was source, target, rating,

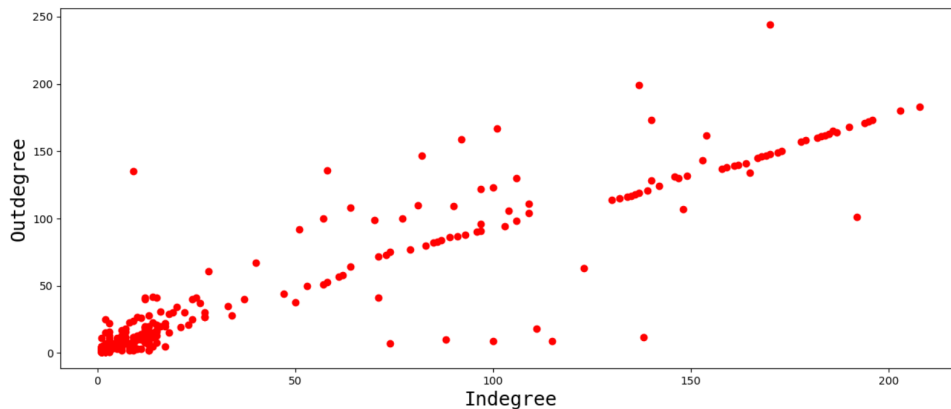


Figure 5.1: Given Vs. Received

and timestamp. All of which is essential information for endorsement network. The direction of endorsement is based on the source and target information. The timestamp is also essential information if using a network anomaly detection algorithm such as net flow rate convergence(cite). The rating shows the strength or weakness of the relation. For the endorsement relation, only the edges with a rating above two are considered.

¹<https://alphabtc.com/blockchain/>

Thus, the total number of nodes considered for endorsement interaction was 1677.

nEG and nER: nEG and nER is outdegree and indegree information respectively. The distribution of which can be seen in figure 5.1. As mentioned in the requirement(req.no), each node needs to have at least 2 or more connections to be considered for impact computation. Failing to do so would result in an impact value equivalent to zero. Of the 1677 nodes in the network, there were 1175 nodes(70%) that had only one incoming or outgoing connection. As a result, only 502 nodes(30%) can have an impact above zero.

Node Label	nEG	nER	ratio	ReceivedPoints	Impact
430	5	3	0.6	0	0
761	2	2	1	0	0
448	3	3	1	0	0
676	2	2	1	0	0
936	2	2	1	0	0

Table 5.1: Nodes with Impact zero because of the receivedpoints

Received Points: The total received point plays an equally significant role in strengthening or weakening the total endorsement impact of a node. This factor relies on the significance of node from whom one receives the endorsement. Table 5.1 shows the nodes that have a maintained ratio of connections but still have zero impact on the network because of the total received points. As can be seen in figure 5.3, the endorser of nodes 430, 761, 448, 676 and 936 has no other connection in the network. Therefore, these nodes have an impact of zero just like their endorsers. The relation of total received point and its contribution to the endorsement impact can be seen in figure 5.2.

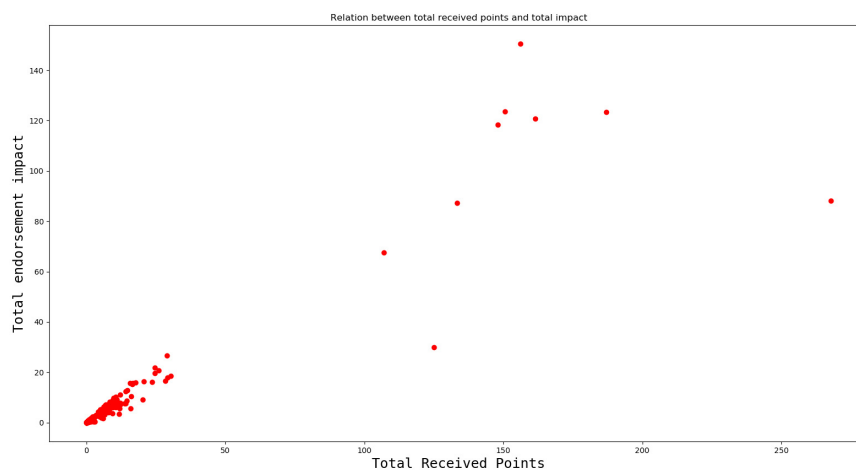


Figure 5.2: Received Points Vs. Total Endorsement Impact

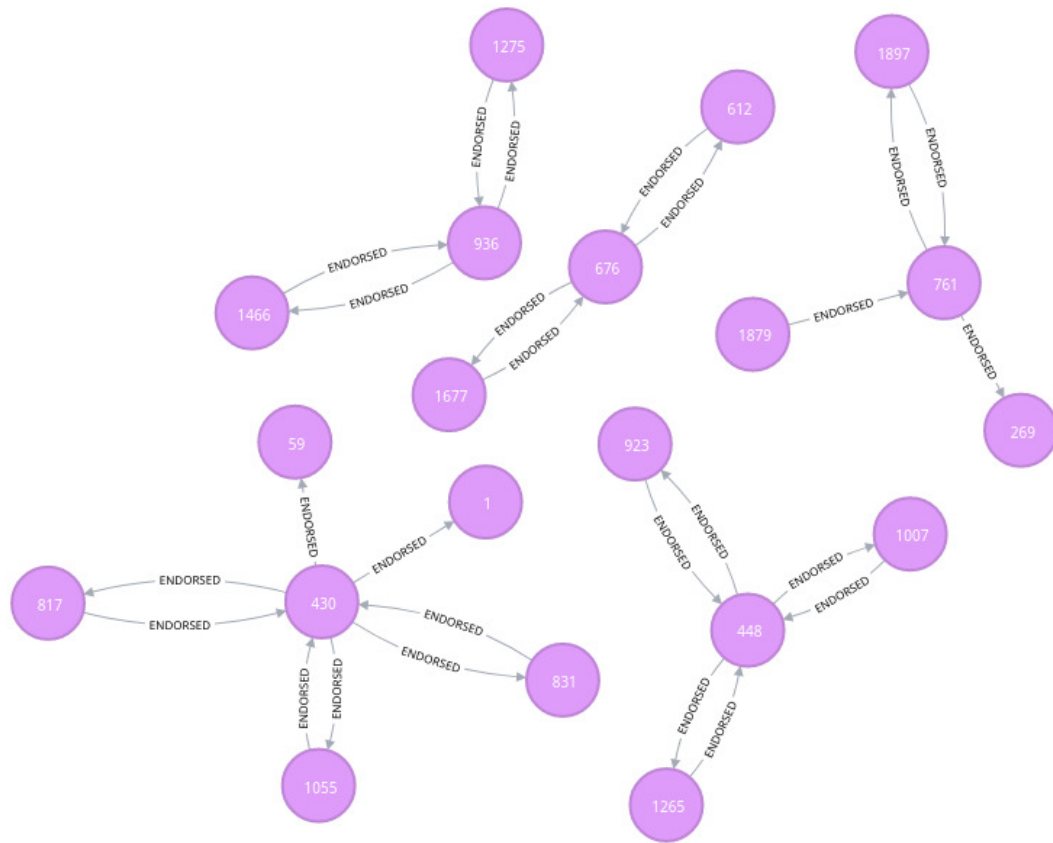


Figure 5.3: Interaction subgraph of nodes with impact zero

Total Endorsement Impact: The endorsement impact in this model alone is affected by the received point and the degree of connections. For the given network, the majority of the nodes have an impact value below 20. There are however few nodes that have managed to obtain an impact value above 100. Each of the nodes having both maximum and minimum impact value is discussed across four scenarios in the section that follows. The distribution of the obtained impact ranges and the number of nodes are presented in table 5.2.

No. of nodes	Impact Range
419	0-5
51	5-10
5	10-15
10	15-20
2	20-25
1	25-30
1	30-35
0	35-65
1	65-70
0	70-85
2	85-90

0	90-115
1	115-120
3	120-125
0	125-150
1	150-155

Table 5.2: No. of nodes and the corresponding impact ranges

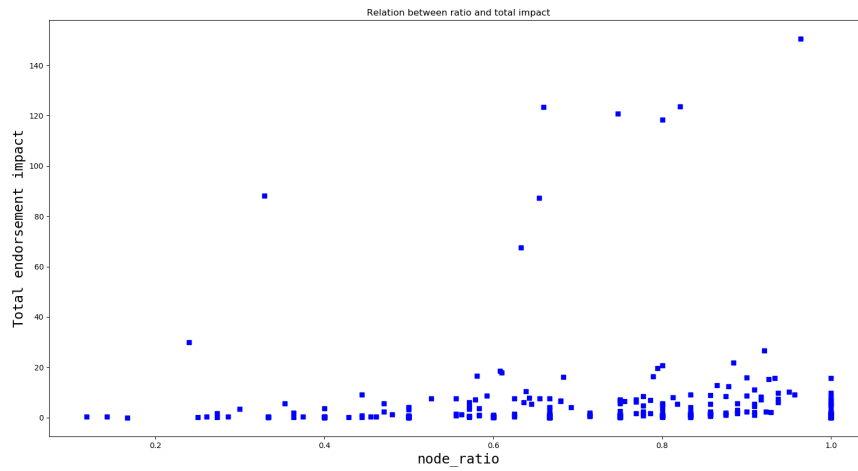


Figure 5.4: Relation of Ratio and Total endorsement impact

5.2 Impact and Ratio: Case-by-case across different scenarios

The representation in table 5.1 and figure 5.4, points that maintaining a ratio of 1 or close to 1 is not enough to have a higher impact on the endorsement network. One can infer that having a good impact implies a good ratio but having a good ratio does not necessarily imply a good impact. As such, a case-by-case analysis for the possible combination of ratio and impact is performed. The maximum value for the impact, ratio, nEG, nER and received points is the maximum value found in the network for the respective variables. The nodes with impact value in the range of 0-15 are taken as low impact nodes, and value above 100 is taken as high impact node for the scenarios.

- **Case1: Maximum Impact/Minimum Ratio:** If a user has been on the network for a while making endorsement connections, then it is possible to have a higher impact. Since receiving an endorsement from many connections cumulative sums up to the existing value. The higher received point, in this case, contributes to the higher impact. The behavior as depicted in figure 5.5 is extracted from the endorsement network among the best impact node with a low ratio.

Case 1: Max Impact, Min Ratio

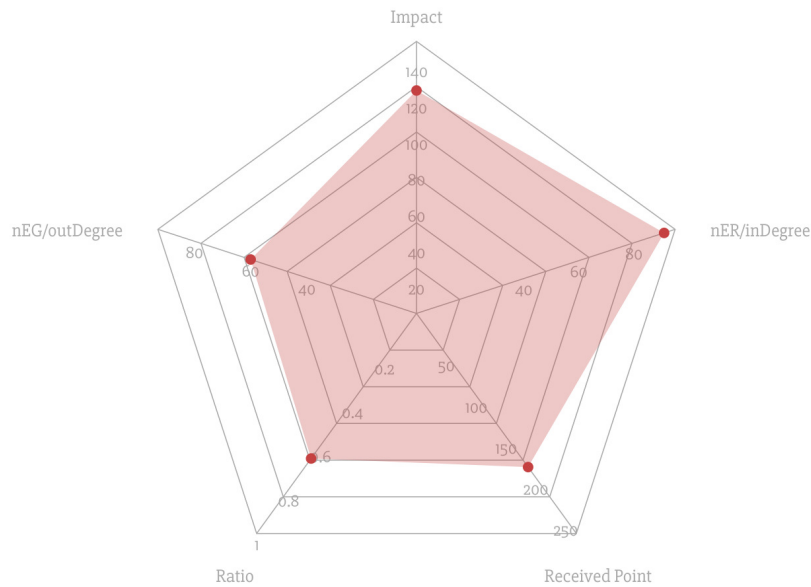


Figure 5.5: Case1- Maximum Impact Minimum Ratio

- Case 2: Maximum Impact/Maximum Ratio: A node with a maximum impact and a maximum ratio is the best case scenario of endorsement network. This scenario implies that a node has been actively participating as both endorser and endorsee. The figure 5.6 shows the behavior of such node. The value across all dimension is almost equally distributed.
- Case 3: Minimum Impact, Minimum Ratio: A minimum impact along with minimum ratio is a sign of node that doesn't have a good trustworthiness range. This case generally appears if the node is only making one-way connections. i.e., giving too many endorsements and receiving none/few or vice-versa. The figure 5.7 shows such behavior.
- Case 4: Minimum Impact, Maximum Ratio: This case has been demonstrated earlier by table 5.1 and figure 5.4. Figure 5.8 gives the behavior of such a node as seen in the network.

Case 2: Max Impact, Max Ratio

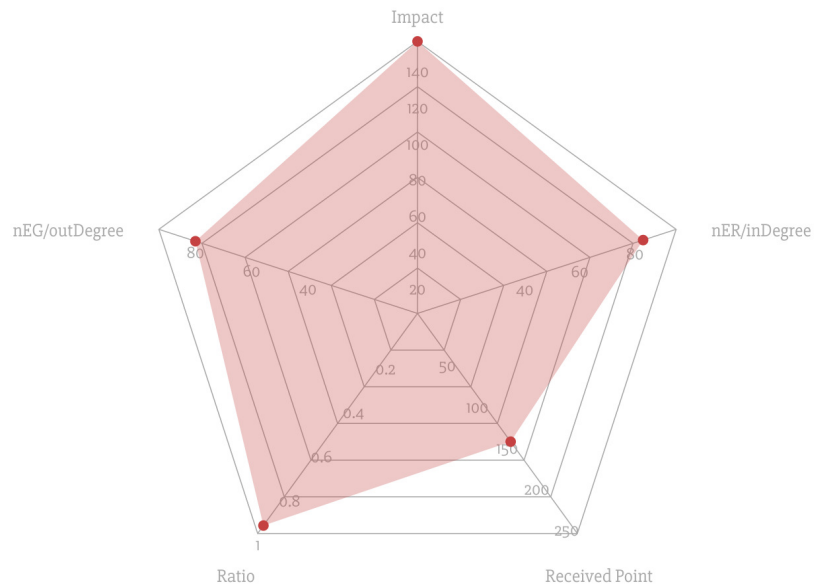


Figure 5.6: Case2- Maximum Impact Maximum Ratio

Case 3: Min Impact, Min Ratio

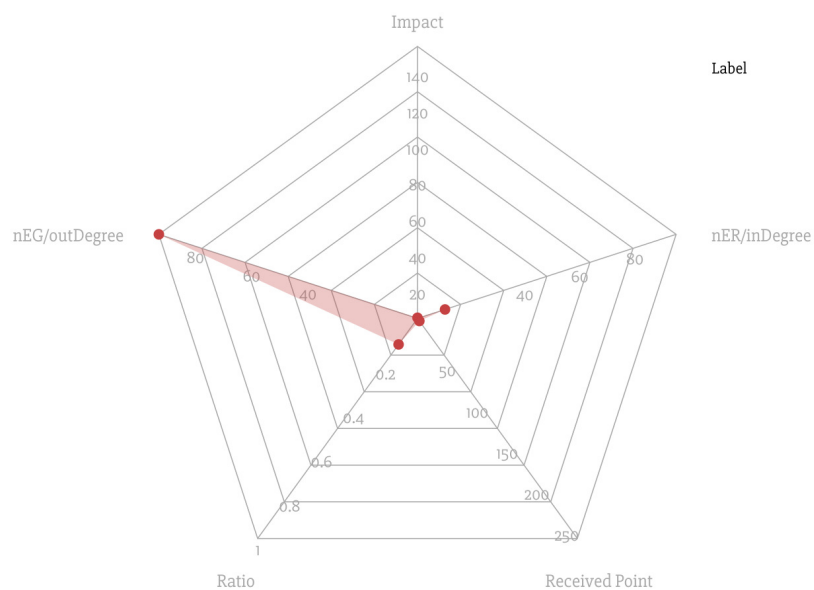


Figure 5.7: Case3- Minimum Impact Minimum Ratio

Case 4: Min Impact, Max Ratio

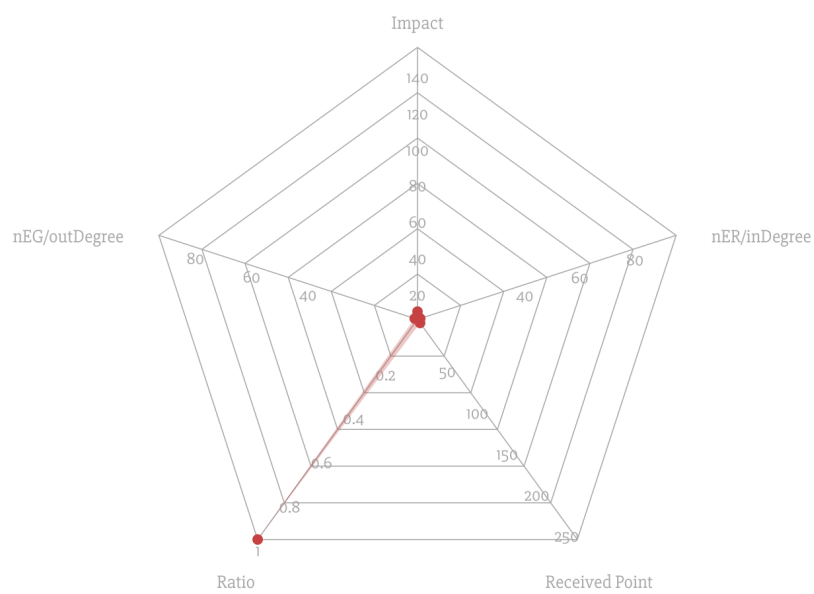


Figure 5.8: Case4- Minimum Impact Maximum Ratio

6 Discussion & Analysis

6.1 Generalization

6.2 first section

7 Conclusion

7.1 first section

7.1.1 first subsection

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] R. Dennis and G. Owen, "Rep on the block: A next generation reputation system based on the blockchain", in *Internet Technology and Secured Transactions (ICITST), 2015 10th International Conference for*, IEEE, 2015, pp. 131–138.
- [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.
- [21] R. Levien, *Advogato's trust metric*, 2003.
- [22] H. A. Kurdi, "Honestpeer: An enhanced eigentrust algorithm for reputation management in p2p systems", *Journal of King Saud University-Computer and Information Sciences*, vol. 27, no. 3, pp. 315–322, 2015.
- [23] S. Alkharji, H. Kurdi, R. Altamimi, and E. Aloboud, "Authenticpeer++: A trust management system for p2p networks", in *European Modelling Symposium (EMS), 2017*, IEEE, 2017, pp. 191–196.
- [24] M. Meulpolder, J. A. Pouwelse, D. H. Epema, and H. J. Sips, "Bartercast: A practical approach to prevent lazy freeriding in p2p networks", in *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, IEEE, 2009, pp. 1–8.
- [25] R. Zhou and K. Hwang, "Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing", *IEEE Transactions on parallel and distributed systems*, vol. 18, no. 4, pp. 460–473, 2007.

- [26] J. A. Bondy, U. S. R. Murty, *et al.*, *Graph theory with applications*. Citeseer, 1976, vol. 290.
- [27] D. Gkorou, “Exploiting graph properties for decentralized reputation systems”, 2014.
- [28] J. Katz, A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 1996.
- [29] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems”, *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
- [30] I. Mironov *et al.*, “Hash functions: Theory, attacks, and applications”, *Microsoft Research, Silicon Valley Campus. Noviembre de*, 2005.
- [31] X. Wang and H. Yu, “How to break md5 and other hash functions”, in *Annual international conference on the theory and applications of cryptographic techniques*, Springer, 2005, pp. 19–35.
- [32] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, “The first collision for full sha-1”, in *Annual International Cryptology Conference*, Springer, 2017, pp. 570–596.
- [33] A. Back *et al.*, “Hashcash-a denial of service counter-measure”, 2002.
- [34] N. Szabo, *Smart contracts*
<http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html>, 1994.
- [35] —, “Smart contracts: Building blocks for digital markets”, *EXTROPY: The Journal of Transhumanist Thought*, (16), 1996.
- [36] —, “Formalizing and securing relationships on public networks”, *First Monday*, vol. 2, no. 9, 1997.
- [37] *What are smart contracts*,
<http://www.chainfrog.com/wp-content/uploads/2017/08/smart-contracts-1.pdf>, ChainFrog, 2017.
- [38] J. Leskovec and A. Krevl, *SNAP Datasets: Stanford large network dataset collection*, <http://snap.stanford.edu/data>, Jun. 2014.
- [39] S. Kumar, F. Spezzano, V. Subrahmanian, and C. Faloutsos, “Edge weight prediction in weighted signed networks”, in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, IEEE, 2016, pp. 221–230.