On Elastic Incentives for Blockchain Oracles

ABSTRACT

A fundamental open question for oracles in blockchain environments is a determination of the amount of trust to be placed in the oracle. Oracles serve as intermediaries between a trusted blockchain environment and the untrusted external environment from where the oracles fetch data. As such, it is important to understand the uncertainty introduced by the oracle in the trusted blockchain environment and the implications of this uncertainty on blockchain performance. This paper develops a model for commoditization of trust. Our model provides for dynamic trust environments that incorporates oracle selfishness. The work also considers the equilibrium behavior for the demand and supply for trust and introduces elastic incentives for increasing the trust. These results are used to determine optimum size of the network that can be served by an oracle with varying degrees of selfishness. Key consequences and challenges of incorporating oracles in trusted distributed ledger environments are presented.

Keywords: Blockchain, Smart Contracts, Demand Function, Elasticity, Blockchain Size

INTRODUCTION

Oracles in blockchain (Wang et al., 2019) function by serving as a bridge between the trusted world of blockchain and the untrusted world outside the blockchain. Although distributed ledger technologies are able to leverage their trust-free architecture to securely process transactions, a major roadblock to their widespread usage is the lack of mechanisms to securely verify and incorporate data that exists outside the blockchain. The inclusion of an oracle helps to bridge this gap between the blockchain and data sources around it. By fetching data from an external source, oracles help to trigger smart contracts that link together the trusted blockchain environment and the untrusted external data source (Wohrer & Zdun, 2018). For example, in a hedging model, nodes in a blockchain (trusted environment) might be dependent on weather data from an external website (untrusted environment) in order to predict future prices for an agricultural commodity. This information cannot be verified using the blockchain's inbuilt architecture for distributed consensus. As a trusted entity, the oracle fetches this information and supplies it to the nodes, thereby triggering a smart contract.

The implicit assumption in existing literature about blockchain oracles is that of an altruistic oracle. This assumption of an altruistic oracle provides for a high degree of trust placed by the blockchain nodes in the oracle's services. However, this assumption of an altruistic oracle may be challenged by factors such as computational complexity of the oracle's tasks and its impact on oracle performance. An oracle that is required to perform computationally intensive tasks to retrieve data from multiple untrusted environments, aggregate and process it for the blockchain is limited by its own computational capacity, the size of the blockchain network that it serves, the quantity of such requests and inter-oracle communication and computation responsibilities. Thus, oracles that are subject to a high volume of computational processing may suffer from degradation in performance in terms of latency, throughput or data accuracy. Additionally, since oracles serve as intermediaries between trusted and untrusted environments, they serve to function as a unique point of failure in the trust model espoused by blockchain environments. An oracle that is manipulated by malware can compromise the integrity of the smart contracts and jeopardize the applications involving such blockchain environments. Such outcomes can then impact the level of trust placed in the oracle by the blockchain and revert the blockchain back to the original state, where the blockchain only trusts data in the ledger and is thus unable to function in hybrid environments that require smart contracts.

Our work studies trust in the institution of the oracle. Specifically, our work seeks to answer the question: How trustworthy is the oracle, and can we use peer evaluations of the oracle's trustworthiness to assess trust placed by a node in the oracle? To do this, we commoditize trust as a tradeable unit with distinct supply and demand functions. Oracles may demonstrate selfish or fair behavior, where selfish behavior behooves the oracle to conserve its own resources and offer subpar service to the nodes. Similarly, a fair oracle is able to serve the requests of the nodes, even at the expense of consumption of its own resources. One reason for an oracle to demonstrate selfish behavior is the amount of work requested of the oracle. The oracle's selfishness is dictated by the number of requests it serves. The role of the oracle's selfishness sets the tone for the number of nodes it can serve. We explore how incentives added to the nodes' trust valuations can influence the number of nodes that can be served by selfish (fair) oracles.

In our model, each blockchain of nodes demands a certain quantity of trust from the oracle, which is quantified by the demand function. The amount of trust demanded from the oracle by the nodes in the blockchain is based on two components, the collective trust placed by peer nodes and the selfishness of the oracle. To do this, first, we borrow upon sociological constructs for trust in societies to formulate a mathematical framework for collective trust. Specifically, we use the definition in Lewis & Weigert (1985) that collective trust in an entity is not just a personal assessment, but it depends on the trust placed by peers in that entity. This sociological conceptualization of trust is extracted largely from the works of Luhman (1979), Barber (1980), Parsons (1963) and Simmel (1900). Luhmann emphasizes that trust leads to a reduction of complexity since individuals can trust their peers to make informed decisions that lessen the amount of risk. Luhmann also states that familiarity is a precondition of trust as well as distrust, as individuals learn and devise strategies for risk-mitigation from the actions and outcomes of their peers. In related work by Simmel (1900), the author environments with perfect information do not impose the need for trust, since individuals would have perfect information about events and outcomes and would know the appropriate strategies to land those outcomes. However, in imperfect environments, trust provides a valuable intermediary construct with which to assess the impact of certain actions and outcomes in the peer environment. Work in Barber (1980) extends this formulation to state that trust implies a "confident expectation" on behalf of the individual that a certain action will result in a less risky outcome. Parsons (1963) goes on to show how trust holds together societies, noting that the lack of trust can erode public confidence and leads to the disintegration of societies.

Our model for trust work as follows. In a network with four nodes A, B, C and D that is asked to trust an entity S, node A might place a level of trust in S that may or may not reflect how its peers ($B, C,$ and D) assess S. In other words, "I trust because they (do not) trust". Additionally, we allow customization of the level of trust placed by every node in the oracle. For example, if nodes $B, C,$ and D together place x units of trust in the oracle, then node A has the option to fully or partially incorporate the collective trust of its peers in its own calculation for the amount of trust it will place in the oracle. Thus, we assume that the collective trust reflects network dynamics of trust and allows for highly scalable trust-enabled

transactions. Second, we assume that the demand function is also dependent on the oracle's selfishness. The demand function is inversely related to the level of the oracle's selfishness, since a blockchain would not highly value the services of a selfish oracle that offers subpar services to the nodes in order to conserve its own resources.

Further, we assume the supply of trust at the oracle as a function of the size of the blockchain it serves and the level of its own selfishness. We assume that as the size of the blockchain increases, the workload of the oracle increases due to the increased number of requests for external data from the nodes. This increased workload has to be accommodated within the limited resources that the oracle possesses. Thus, the oracle is required to "do more with less", thereby affecting the network performance. We model this assumption of an inverse relationship between network size and oracle performance in terms of the commoditized trust variable. An oracle that is able to perform subpar work is able to serve more nodes than when it consistently performs at the highest level of computational integrity. In essence, the increase in size of the blockchain results in a decrease of trust that the oracle can supply. We further provide bounds on the oracle's performance by modeling it as an agent with preferences toward serving the blockchain. The motivation for this model is in Sliwka (2007), where the authors study the economics of incentives in a setting with three types of agents: selfish agents, fair agents and conformist agents. While the first two types of agents behave in selfish/fair manners, the conformists are motivated to act fairly only if they think that a sufficient number of other agents are behaving fairly as well. We apply this assumption of a continuum of selfishness to our system of the blockchain nodes and the oracle, where a node formulates its trust in the oracle based on the trust placed in the oracle by its peers. We assume that a selfish oracle exhibits high degree of incompetence to accomplish a task so that task accomplishment utilizes the least amount of its computational resources. On the other hand, we assume that a fair oracle seeks to maximize the utility of the blockchain by exhibiting least amount of incompetence. We study the interaction between collective trust, degree of oracle's selfishness and the size of the network that a selfish or fair oracle can serve.

Finally, we study the sensitivity of the demand function of trust to changes in the collective trust placed by the blockchain in the oracle. We define a variable, called the **collective trust elasticity** (CTE), that measures the sensitivity of the demand for trust from the blockchain as function of changes in collective trust. The motivation for measuring the elasticity of the demand for trust lies in its scarcity in the blockchain-oracle system. The elasticity of demand functions has its origins in microeconomics (Anderson et al., 2010) where the demand for a good is assessed relative to its price. Consequently, elasticity of demand for various commodities across several domains has been studied, including crude oil (Cooper, 2003), electricity (Kirschen et al., 2000) and patents (Rossenfosse & Potterie, 2012). We use the same notion of elasticity to measure the impact of varying collective trust and oracle selfishness parameters on the CTE. The CTE is calculated within the bounds of the degree of selfishness of the oracle. We use these results for the CTE to inform the design of a blockchain where the oracle was incentivized to offer a consistent level of service to the blockchain. Thus, instead of an oracle that choses to behave in a selfish or fair manner, we offer an incentive to the oracle to perform with reasonable level of altruism. The implementation of such incentives can take various forms, such as increase in computational resources, decrease in network size or number of requests. We develop a mathematical model for the incentive and analyze the behavior of the oracle and the blockchain at equilibrium for the case of an incentivized oracle. These results are then used to determine the optimum number of the nodes that can be served by an oracle in both selfish and fair modes of operation.

To the best of our knowledge, our work is the first study to systematically explore the consequences of misplaced trust in the oracle. We broadly use the term ‘oracle’ to refer to both centralized and decentralized oracles. Additionally, our work can be applied to both software and hardware oracles that serve as intermediaries between the blockchain and untrusted environments. Our work has implications in areas where smart oracles, powered by deep-learning algorithms, can choose to offer varying Service Level Agreements (SLAs) to the blockchain. These deep-learning algorithms could infer key metrics about the blockchain and create smart contracts that dynamically create incentives for optimizing network performance. Thus, the oracle’s selfishness and trust placed by nodes in the oracle will significantly

impact the network performance. This is especially true for blockchains with multiple oracles, where the performance of oracles will vary according to their load and self-preserving motives.

BACKGROUND

The fundamental question that this paper seeks to address is that of trust in an oracle's services. In this section, we showcase the notion of trust in three contexts: (a) sociological constructs of trust in traditional, offline networks, (b) trust in peer-to-peer systems and crowdsourced systems, and (c) trust in blockchain systems and oracles. These three contexts follow the natural evolution of trust-based applications, moving from offline networks to peer networks to the blockchain.

Trust in offline social networks: The notion of trust has been studied through the lens of computational sociology focusing on aspects such as reputation and credibility. In Benabou &Tirole (2016), the authors develop a theory of individual's prosocial behavior as a function of the individual's concern for reputation. Their model showed that extrinsic incentives offered for good deeds had the potential of spoiling the reputational values of the deed, since it could be postulated that the motive of the good deed was inspired by the incentive. They also showed how prominence and memorability were strong incentives in encouraging prosocial behavior, and greater reputation might lead to lesser commitment from individuals who want to refrain from overtly disclosing their deeds. Further, the model showed that a person's actions might influence others to take up actions that are either strategic substitutes or complements. The former occurs when the actions generate negative publicity, while the latter occurs in instances of positive publicity. Finally, the authors showed that incentives for prosocial behavior can affect the elasticity of the supply curve. Gachter et al. (2013) studies the influence of peer effects in shaping an individual's preferences and actions. Further evidence of the incentives in the form of social diversity serving to encourage prosocial behavior is in Kovarik et al (2012). Here, the authors show that the behavior of a few central individuals are readily propagated through the network, and these roles are occupied by more prosocial individuals. The role of trust in virtual communities has been studied in Nah et al. (2017), where the authors study the impact of task complexity on trust and show that trust is the

intermediary between task complexity and team satisfaction. Additional work on the role of trust in teams has been documented in Siau et al (2010) where the authors explore the role of trust in team projects.
A detailed exposition of the interplay between provenance, trust and reputation is in Packer et al. (2014) which informs their design of an auditable reputation service. Here, the authors describe four categories of use cases in reputation and provenance: functional, auditable, privacy/security and administrative. A high reputation impacts the trust that users and peers place in the system, thereby promoting accountability. The work in Macy & Skvoretz (1998) studies the evolution of trust between strangers. The authors study trust in two types of interactions – those between neighbors and those between strangers. Using genetic algorithms to study the evolution of trust and cooperation in both scenarios, the study found that trust evolves in stages – initial widespread distrust, followed by drift toward trustworthiness, local trust, local cooperation, diffusion via contact with strangers and finally culminating in universal trust. They found that the spread of cooperation and trust was heavily influenced by social proximity and in-group altruism. Work in Mui et al. (2002) provides a brief survey of trust and reputation models and then proposes a computational model that incorporates the role of reciprocity and reputation in building trust. This work structured reputation as a function of the social network that the agent is embedded in, thus leading to trust being derived from reputation. The role of a trust network in knowledge-sharing in multi-agent societies was studied in Ding et al (2004). Here, trust was determined from the contributions of an agent to a knowledge-base. They showed that trust was crucial in the propagation of information through a combination of strong and weak trust paths. Thus, the length of a propagation path is used as an indicator of trust, where longer paths are considered more trustworthy. Finally, trust also played into data retrieval in determining the most trustful set of agents to be queried for retrieval of truthful information. The role of trust in forming and maintaining social relationships was studied in Sutcliffe & Wang (2012). This study proposed that when a history of interactions is available, social relationships consisting of strong, medium and weak ties evolve through the presence of an affect component. Strong ties are sustained despite diminishing or plateaued utilities, suggesting that emotional constructs rather than returns govern the sustenance of the relationship. In contrast, medium and weak ties create scope for a

higher degree of non-cooperative behavior that does not reflect levels of deep trust. Another computational model of trust was studied in Czap (2005). The authors modeled trust as a dynamic factor that varied as a function of the following factors: agent's trust in a peer, the weight attached to the trust relative to profitability and the agent's own trustworthiness modeled as a threshold of defection. They showed that in general, trustful multiagent systems tend to be more profitable. The model also showed that over time, cooperation and discovery influenced the degree of trust in a relationship. The role of trust in information disclosure was surveyed in Nixon et al. (2005), where the authors study the impact of security, privacy and trust in smart environments for pervasive computing.

Trust in peer-peer and crowdsourced systems: The role of trust in a sharing economy has been studied in Hawlitschek et al. (2016), where the authors propose a model for trust that is differentiated among three categories of people, platform and product. The authors showed that higher level of trust in the three categories positively influence decisions to consume the product. The work in Yu et al. (2012) studies the notion of trust in crowdsourced systems. They showed that designing crowdsourced systems that rely solely on trustworthiness of the individuals might result in decline of social welfare, since they observed that the natural propensity of individuals is to cheat when given a chance. The authors proposed a trust management model that enhances the system by focusing on both fair treatment of trustworthy workers and increase in the social welfare of the system. Trustworthiness of crowd-reported information in crowdsourced systems has been studied in Venanzi et al. (2013). Here, users' trustworthiness is estimated from the fused outputs of the crowd through a model that is also sensitive to collusion attacks. In Wang et al. (2016), the authors explore crowdsourcing trust in the paradigm of the social Internet of Things (SIoT), where humans and devices relay on each other for services. The trust is introduced in the form of a reputation-based auction mechanism, where trustworthy participants are selected and awarded incentives for completing tasks in a trustworthy manner. The role of trust in motivating individuals to participate in crowdsourcing was also explored in Ye & Kankanhalli (2017), where the authors develop a model for crowdsourcing participation based on benefits, costs and trust. They found that among other factors, trust increased the participation and trust was positively affected by rewards.

A role-based trust model for trust management in peer-peer systems was proposed in Khambatti et al. (2004). Here, the trust values are assigned proportional to peer status. Thus, a peer with more links with higher link weights is considered more trustworthy. Likewise, peers can be punished for malicious behavior by severing of links causing a decrease in trust and social status. An analysis of the attributes of trust is in Bhattacharya et al. (1998), where the authors describe trust as a desirable entity that exists in uncertain and risky environments, is probabilistic and is situation and person-specific. The authors provide several examples of trust-inducing interventions including incentives, peer pressure and strategic network design. The importance of trust in building recommender systems was explored in Medo et al. (2009) for the specific application of online news recommendation. The model reflects the opinions of like-minded individuals and prior submission records to determine best authorities for news recommendations. Issues of trust in crowdsourcing applications were studied in Weaver et al. (2012) for specific applications of crowdsourcing real-time information in the aftermath of disasters. The authors investigated trustworthiness of submitted information, where malicious actors like terrorist groups might submit untrustworthy information to divert disaster-response resources. They explore trustworthiness as a function of three tools, in increasing order of sophistication. In the first tool, trust is determined by the level of trust in the group that the individual belongs to. For e.g., a user belonging to the group of admins might possess a higher level of trust than an anonymous user. In the second tool, trust is determined by a crowdsourcing application where user click on stars (0-5) to indicate their rating of data posted by the user. Over time, these ratings are aggregated to indicate trustworthiness and usefulness of the data. The last tool for enhancing trust is in the form of machine-learning algorithms that are harnessed to combine multiple sources of data in determining the truthfulness of posted data. For e.g., a user who reports an earthquake will generate a higher trust rating in the model if the data can be verified from USGS earthquake reporting tool. The notion of incentives to increase trust has been studied in Goel et al. (2019), where authors study scenarios in which questions are to be answered on a decentralized platform. Agents are rewarded for their contributions to the outcomes of the questions. The model showed that even with

the presence of a fraction of honest agents, undesirable equilibria (non-truthful resolution of the questions) can be averted in favor of honest outcomes.

Trust in blockchain, smart contracts and oracles: An overview of trust mechanisms in decentralized blockchain-based markets in Subramanian (2018). Here, the author advocates the use of reputation programs that utilize the decentralized, immutable nature of the blockchain to provide trustworthy services in a transaction market. The use of the blockchain has multiple advantages including a decrease in transaction time and affords the implementation of smart contracts which results in lower transaction costs, and increased privacy and security. The foundations of cryptographic trust in blockchain were leveraged in Zou et al. (2018) to develop a new consensus protocol called Proof-of-Trust (PoT), which can be leveraged for use in crowdsourcing applications. Similar to blockchain Proof-of Work (PoW), the PoT consensus protocol chooses transaction validators based on the trust value of participants.

A systematic study of smart contract architectures is in Wang et al. (2018), and specific applications to the lifecycle of a contract (formation and negotiation, storage, enforcement, monitoring, modification and dispute resolution) is presented in Idelberger et al. (2016). Work in Beck et al. (2016) discusses the concept of trust in a smart contract and illustrates it with the help of a smart contract designed for intermediary-free transactions. In Watanabe et al. (2016), the authors study the concept of credibility in a blockchain contract, where credibility is used as a core component of trust. In this work, a higher number of contracts implies a higher level of trust. This credibility-based model has a two-fold objective: enhancing the trust that peers place in the node, and serving as a deterrent for malicious behavior for fear of negative credibility among peers. The use of smart contracts for insurance markets was explored in Sheth and Subramanian (2019), where the blockchain was used to automate insurance transactions and record critical data in pricing, underwriting, issuance and claim settlements in a trusted environment.

Decentralized oracles were first proposed in Peterson et al. (2015). The authors proposed Augur, a decentralized oracle that allows people to trade in prediction markets for low costs. In Augur, the ability to create a market is available freely, and all users may freely trade on any market. The system uses reputation tokens as a reward/penalty mechanism to regulate the reporting of the market outcomes and its

distance from the outcomes of the actual event. Honest reporting of outcomes was shown to result in stability of the prediction markets using incentives that increased with the number of reputation tokens amassed for the reporting. The incentive structure developed in Augur seeks to increase trust by leveraging reputation. Additional incentives are made available in the form of fees and bonds that are paid out to market creators and reporters. The work in Freeman et al. (2017) studies a mechanism for increasing truthful reporting in prediction markets. In this mechanism, truthfulness of reporting is enhanced by basing the outcome of prediction on the proportion of arbiters who vote affirmatively for the outcome, instead of a binary outcome. Peer arbiters can increase the strength of a signal based on the signals received by peers. Additionally, arbiters are rewarded for their efforts in gathering and reporting information. These rewards are funded by trading fees imposed on the agents. The role of an oracle in prediction markets was also studied in Adler et al. (2018), where the work studies ASTREA, a decentralized oracle that attests to the veracity of Boolean propositions. The authors use a game-theoretic framework with three players: voters, submitters and certifiers. Submitters submit a Boolean proposition to the system and pay a fee for the services. Voters play a low-risk game by providing Boolean answers to the question. Certifiers play a high-risk game where they place larger stakes on the outcomes of the voting and certification process. The framework is designed with rewards/penalties for the players, depending on the way they vote. They show that this framework achieves a desirable Nash equilibrium where all rational players behave honestly. Reputation-powered incentives were also studied in dePedro et al. (2017) where the authors propose Witnet, a framework where miners earn tokens called Wit for retrieving, accessing and delivering data from websites to the blockchain. An authenticated data feed system called TownCrier that incorporates confidentiality was proposed in Zhang et al. (2016). This work proposed an end-end solution for implementation of Town Crier that used trusted computing platforms to guard against malicious processes. The question of a dysfunctional oracle was raised in Allen et al. (2019), where the authors pondered upon the implications of a malfunctioning oracle on smart contract performance.

Our work leverages the altruistic preferences of the oracle and the trusting preferences of the blockchain to dynamically assess the trustworthiness of the oracle. Thus, the smart contracts generated between the oracle and blockchain are informed by a behavioral contract theory approach (Koszegi, 2014). Further work on the generation on tamper-proof random numbers for smart contracts on the blockchain is in Chatterjee et al. (2019) where the authors call attention to the dearth of truly random numbers, thus rendering the blockchain vulnerable to tampering by miners. Additional work on security of smart contracts is in Nehai et al. (2018), Magazzeni et al (2017) and Tsankov et al. (2019). Specifically, our work differs from existing work in blockchain oracles in two ways. First, we assume that the oracle is able to choose its altruistic preference toward servicing the blockchain. Second, we question the heretofore held assumption of complete trust placed in the oracle by the blockchain. The implications of these assumptions extend to situations where competing parties place varying amounts of trust and have different world-views about the outcomes of a market. Thus, our model builds a model for trust that is decentralized and dynamic. This notion of trust placed in the oracle by the blockchain is unlike the trust required of workers or peers in crowdsourcing systems. Traditional crowdsourcing systems involve the use of "crowds" to provide services in a dynamic manner, whereas the blockchain oracle is a single entity that provides a requested service to the blockchain. Thus, instead of assessing the trustworthiness of a large set of nodes as in traditional crowdsourced systems, our work studies the trustworthiness of the oracle based on the trustworthiness ascribed to the oracle by fellow nodes. In keeping with the decentralized nature of blockchain, our model for trust reflects individual assessment of peer trust, as well the node's assessment of the oracle's selfishness. Unlike previous work that assumes honest agents, our work differs in that we do not assume the existence of honest agents. Instead, trust is built over time as a function of self and peer evaluations of an oracle, and even then, our model gives the agents free will to act rationally or irrationally by refusing to follow the wisdom of the peer network. Thus, unlike the previous work in Peterson et al. (2015), or Adler et al. (2018) that assumes the use of non-malicious resolution sources or honest reporters, our work allows for dynamic updates of trust in the oracle's services based on peer assessment of the oracle's work. Additionally, to combat the influence of overtly

malicious peers, a node may choose to only partially use the collective trust values of its peers in its own formulation for trust.

COMMODITIZED TRUST MODEL FOR THE ORACLE-BLOCKCHAIN SYSTEM

Demand Function for Nodes

Since the notion of an oracle is built on the premise of trust, our model commoditizes trust so that there is market for trading trust. The nodes in the blockchain demand a value of trust from the oracle, specified by the demand function D . We model the demand function D as dependent on two parameters – we assume that the parameters for the demand function are the collective trust placed by all nodes in the network (c_{coll}) and the selfishness of the oracle θ. Thus, $D = f(c_{coll}, \theta)$, signifies the demand for trust from the nodes in the system.

For some constants ω_1 and ω_2, we can write D as

$$D = \omega_1 c_{coll} - \omega_2 \theta \qquad (1)$$

The term for selfishness θ has a negative slope, since the higher the level of oracle's selfishness, the lower is the amount of trust demanded the blockchain. Similarly, the higher the level of the collective trust c_{coll}, the higher is the demand for trust from the blockchain. Table 1 shows the variables in our model.

Table 1. List of symbols and their meanings

Variable	Meaning
D	Demand function for trust
c_{coll}	Collective trust
c_i	Trust placed in oracle by node i
θ	Degree of oracle selfishness
ω_1, ω_2	Constants in demand function Q_d

α_i	Trust modulation factor for node i
S	Supply function for trust
β_1, β_2	Constants in supply function Q^s
n	Number of nodes served by oracle
D_{eq}	Demand function at equilibrium
E	Collective trust elasticity
ρ	Percentage of nodes that receive incentives
μ	Amount of incentives added to individual node's trust values
$D_{incentive}$	Demand function for trust with incentives
n_s	Number of nodes supported by selfish oracle
n_f	Number of nodes supported by fair oracle

We now develop the framework for modelling the collective trust. For a network of n nodes on the blockchain, we assume that the collective trust dynamically reflects the amount of trust that a node i places in the network plus the impact of other nodes' trust in the oracle. To accommodate varying preferences for dependence on social norms for trust, every node does not fully change its trust in the oracle to suit the trust preferences of other nodes in the networks. Each node modulates the trust of its peers by the trust modulation factor, denoted by α_i, where $0 < \alpha < 1$. If $\alpha = 0$, it implies that node i does not trust its peers' assessment of the of trust placed in the oracle. A value of $\alpha = 1$ implies that the

node i fully trusts its peers' assessment of trust placed in the oracle. The following equation summarizes this model of collective trust:

$$c_{coll} = \sum_{i=1}^{n} \left(c_i + \sum_{j=1}^{n} \alpha_i \frac{c_j - c_i}{n-1} \right) \quad (2)$$

where $j \neq i$.

To illustrate this framework (Figure 1), consider nodes A, B, C and D that are placing their trust in oracle O. The amount of trust placed by the nodes in O are in Table 2.

The nodes can pass a hard information signal similar to work in Rajan, et al. (2010) containing their trust values. Nodes also have their own belief system to assign a level of trust in their peers' trust assessment of the oracle. This belief system is quantified by the variable α_i, which denotes the level of trust that node i places in the trust values of other nodes in the system. For the system described in this example, the nodes can update their trust value as follows.

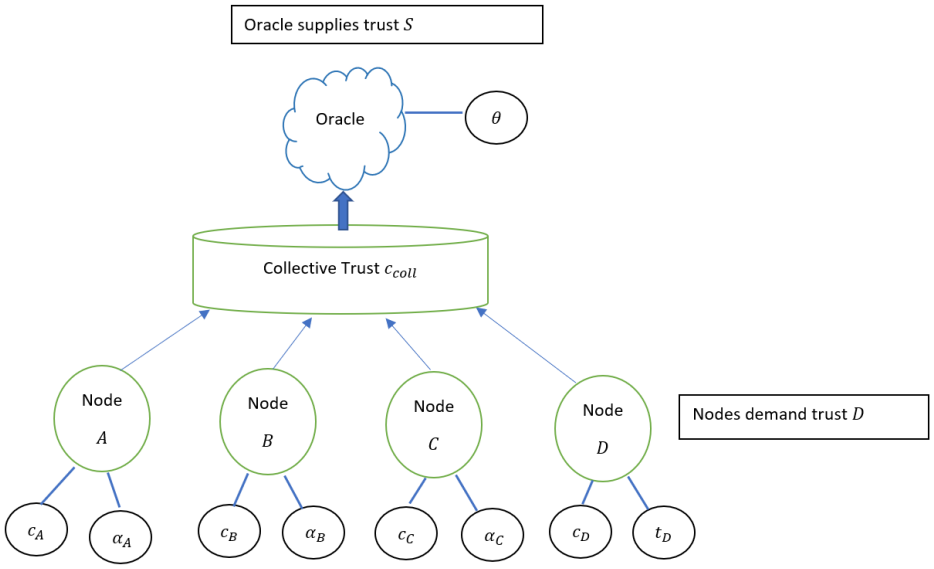

Figure 1: A blockchain system of an oracle and four nodes (A, B, C, D). *Each node has two attributes, the trust it places in the services of the oracle* c_A, c_B, c_C, c_D) *and the trust it places in the trust placed by its peers in the oracle's services* $(\alpha_A, \alpha_B, \alpha_C, \alpha_D)$. *The collective trust* c_{coll} *placed by all nodes in the oracle*

and the oracle's selfishness θ determine the demand function D. The oracle supplies trust S as a function of its selfishness and the number of nodes n.

Table 2. Example of trust assignments for computation of collective trust c_{coll}

Node	Trust Value
A	5
B	6
C	3
D	8

For node A:

$$c_1 = 5 + \alpha_1 \frac{(6-5)+(3-5)+(8-5)}{3} = 5 + \frac{2\alpha_1}{3}$$

For node B:

$$c_2 = 6 + \alpha_2 \frac{(5-6)+(3-6)+(8-6)}{3} = 6 - \frac{2\alpha_2}{3}$$

For node C:

$$c_3 = 3 + \alpha_3 \frac{(5-3)+(6-3)+(8-3)}{3} = 3 + \frac{10\alpha_3}{3}$$

For node D:

$$c_4 = 8 + \alpha_4 \frac{(5-8)+(6-8)+(3-8)}{3} = 8 - \frac{10\alpha_4}{3}$$

Thus, the collective trust c_coll is given by

$$c_1 + c_2 + c_3 + c_4 = 22 + \frac{2\alpha_1}{3} - \frac{2\alpha_2}{3} + \frac{10\alpha_3}{3} - \frac{10\alpha_4}{3}$$

Simplifying equation (2) for a system of n nodes, we get

$$c_{coll} = \sum_{i=1}^{n} c_i(1-\alpha_i) + \frac{1}{n-1}\sum_{i=1}^{n}\left(\alpha_i \sum_{j=1}^{n} c_j\right) \ , j \neq i \qquad (3)$$

Substituting equation (3) in equation (2) for the demand function for trust, we get

$$D = \omega_1 \left[\sum_{i=1}^{n} c_i(1-\alpha_i) + \frac{1}{n-1}\sum_{i=1}^{n}\left(\alpha_i \sum_{j=1}^{n} c_j\right)\right] - \omega_2\theta \qquad (4)$$

Supply Function at the Oracle

Next, we study the supply function for trust S. The amount of trust that the oracle O can supply to the nodes is given by

$$S = -\beta_1 n + \beta_2\theta \qquad (5)$$

Equation (5) reproduces the impact of resource constraints (computational capacity, performance, throughput) of the oracle on the supply of trust. We assume that the selfishness of the oracle is inversely proportional to the amount of trustworthy work done by the oracle. The inverse relationship between the size of the network and its selfishness is driven by the increased workload that the oracle has to satisfy for a larger network. Examples of this inverse relationship between size and performance are manifested in blockchain networks (Scherer, 2017), and are widespread in diverse settings such as wireless networks (Gupta & Kumar, 2000), wireless sensor networks (Tian & Georganas, 2002) and large ad hoc networks (Naumov & Gross, 2006). The negative slope of the supply function S with respect to the size of the blockchain n indicates that the as the size of the blockchain increases, the supply of trust at the oracle decreases. However, as the size of the blockchain increases, its selfishness increases. This is reflected in the positive slope of the supply function for trust with respect to the selfishness.

The rest of this analysis assumes the following two tendencies of oracles:

- **Selfish oracle:** A selfish oracle demonstrates a high propensity to conserve resources in the face of increasing demands from a larger network of nodes, i.e. $\theta \rightarrow 1$. Thus, a selfish oracle would exhibit a value of θ closer to 1.

- **Fair oracle:** A fair oracle minimizes the propensity to conserve resources and is fair to the nodes by providing higher quality of service to the nodes, i.e. $\theta \to 0$. Thus, a fair oracle would exhibit a value of θ closer to 0.

It is important to note a clarification in this categorization of oracles. Although we deem oracles as fair or selfish, the value of the degree of oracle selfishness, θ, lies along a spectrum from 0 to 1. Thus, given two different values of selfishness, θ_1 and θ_2, if $\theta_2 > \theta_1$, then the oracle with selfishness value θ_1 is considered more selfish even if θ_1 tends to 1.

Equilibrium Analysis of Demand and Supply Function for Trust at the Oracle

We now model the equilibrium behavior for the demand and supply functions of the oracle and the blockchain. Equating equations (4) and (5), we get

$$D = S$$

$$\omega_1 t_c - \omega_2 \theta = -\beta_1 n + \beta_2 \theta \qquad (6)$$

Selfish Oracle

For a selfish oracle, the value of the selfishness parameter θ tends closer to 1. Equation (6) thus simplifies to

$$c_{coll} = \frac{\beta_2 - \beta_1 n + \omega_2}{n} \qquad (7)$$

Substituting (7) in equation (4), we get the demand function, D_{eq},for a selfish oracle at equilibrium.

$$D_{eq} = \beta_2 - \beta_1 n \qquad (8)$$

Fair Oracle

For a fair oracle, the value of the selfishness parameter θ is closer to 0 ($\theta \to 0$). In this case, then, equation (6) simplifies to

$$c_{coll} = \frac{-\beta_1 n}{\omega_1} \quad (9)$$

Substituting equation (9) in (4), we get the demand function, D_{eq} for a fair oracle at equilibrium.

$$D_{eq} = -\beta_1 n \quad (10)$$

Elasticity of Collective Trust

Since collective trust placed by the blockchain in the oracle is a significant component of the demand function, we study the sensitivity of the demand function to the collective trust. Similar to the previously mentioned notion of elasticity of demand functions with origins in microeconomics (Anderson et al., 2010) where the demand for a good is assessed relative to its price, we define the Collective Trust Elasticity (CTE), denoted by E. Thus, the CTE, E, is given by the sensitivity of the demand function to changes in the collective trust

$$E = \left(\frac{\Delta D}{\Delta c_{coll}}\right)\left(\frac{c_{coll}}{D}\right) \quad (11)$$

Substituting equation (1) in (11), we get

$$E = \frac{\omega_1 c_{coll}}{\omega_1 c_{coll} - \omega_2 \theta} \quad (12)$$

Equation (12) shows that the CTE is bound by the selfishness of the oracle.

Selfish Oracle

As the value of the selfishness parameter θ tends closer to 1, equation (12) yields

$$E = \frac{1}{1 - \omega_2/(\omega_1 c_{coll})} \quad (13)$$

Equation (13) shows that when an oracle gets more selfish, as the collective trust c_{coll} increases, the sensitivity of the demand function to the collective trust decreases. This is an example of an elastic demand function that demonstrates the impact of collective trust on the CTE value.

Fair Oracle

As the value of the selfishness parameter θ tends closer to 0, equation (12) yields

$$E = 1 \tag{14}$$

Equation (14) denotes a perfectly elastic function. An increase in the value of the collective trust does not increase the demand for trust, thereby showing that the network prefers a fair oracle over a selfish oracle.

Bounds on the Collective Trust Elasticity

We now study the bounds of E by using the oracle's tendency toward selfishness or fairness as the limits. Combining equations (13) and (14), we get the bounds for E.

$$1 \leq E \leq \frac{1}{1-\omega_2/(\omega_1 c_{coll})} \tag{15}$$

Creating Incentives

Incentives have been shown to alter the environment (Bowles & Polania-Reyes, 2014) in which new preferences are learned, and our model aims to enhance the performance of the oracle-blockchain smart contract system using incentives. We envision incentives such as allocation of additional computational resources, or the creation of locked contracts that guarantee a fixed fairness and load for specific durations. Since incentive design has been shown (Benabou & Tirole, 2006) to be a significant tool in creating more efficient markets, a blockchain system that can leverage the utility of incentives can create self-regulating environments for use in distributed ledger applications.

We assume a reasonable oracle ($\theta = 0.5$) whose level of selfishness is at the midpoint of complete selfishness ($\theta = 1$) and complete fairness ($\theta = 0$). We create an elastic incentive for nodes, based on the CTE, to increase their trust in the oracle based on the assumption of a reasonable oracle. We further model the incentives as non-uniformly applied, i.e. a percentage ρ of the nodes elect to receive incentives, whereas others do not. The amount of incentive added to an individual node's trust is obtained from the midpoint of the bounds specified by equation (17). Thus, the incentive μ, is given by

$$\mu = \frac{1}{2}\left[1 + \frac{1}{1-\omega_2/(\omega_1 t_c)}\right] \quad (16)$$

The demand function with incentives, denoted by $D_{incentives}$, is formulated by adding the incentive μ to $\rho\%$ of nodes as follows:

$$D_{incentive} = \omega_1 t_c + \frac{\rho n}{2}\frac{1}{2}\left[1 + \frac{1}{1-\omega_2/(\omega_1 t_c)}\right] - \omega_2\theta \quad (17)$$

Here, we study how incentives modify the behavior of the system at equilibrium for the demand and supply functions for trust.

$$D_{incentive} = S \quad (18)$$

Substituting equations (5) for the supply function and (17) for the CTE in (18), we get

$$\omega_1 t_c + \frac{\rho n}{2}\frac{1}{2}\left[1 + \frac{1}{1-\omega_2/(\omega_1 t_c)}\right] - \omega_2\theta_I = -\beta_1 n + \beta_2\theta \quad (19)$$

Optimal Number of Nodes in the Incentive Model

We are interested in finding the optimal number of nodes that can be served by an oracle where $\rho\%$ of nodes are offered incentives. To find this, we optimize equation (21) for the equilibrium behavior with incentives.

Let $f(c_{coll}, n) = \omega_1 t_c + \frac{\rho n}{2}\frac{1}{2}\left[1 + \frac{1}{1-\omega_2/(\omega_1 t_c)}\right] - \omega_2\theta_I + \beta_1 n - \beta_2\theta$ (20)

We optimize equation (22) for $\theta = 1$ to find the number of nodes n_s that can be supported by a selfish oracle. Setting the partial derivatives of f with respect to c_{coll} to be equal to zero, we get,

$$\frac{\partial f}{\partial c_{coll}} = 0 \quad (21)$$

For $\theta_I = 1$, equation (21) results in

$$c_{coll} = \frac{\omega_2}{2\omega_1} - \frac{\rho n}{4\omega_1}$$

Setting the partial derivatives of f with respect to n to be equal to zero, we get,

$$\frac{\partial f}{\partial n} = 0 \quad (22)$$

For $\theta_I = 0$, equation (21) results in

$$c_{coll} = \frac{\omega_2}{2\omega_1} - \frac{2\beta_1}{\rho\omega_1} \qquad (23)$$

Equation (22) and (23), we get the number of nodes that can be supported by a selfish oracle (n_s) as follows:

$$n_s = \frac{8\beta_1}{\rho^2} \qquad (24)$$

Repeating the analysis in equations (21) – (23) for the case of a fair oracle $(\theta_I = 0)$, we get the number of nodes supported by a fair oracle (n_f) as follows:

$$n_f = \frac{2\beta_1}{\rho^2} \qquad (25)$$

Comparing equations (24) and (26), we see that the oracle can support a quarter of the nodes in fair mode than when operating in the selfish mode.

RESULTS AND ANALYSIS

In this section, we evaluate the performance of the oracle and the blockchain environment with our model of commoditized trust. We present the simulation results of four components of our model – (i) CTE of the trust demand function with varying levels of the selfishness parameter θ, including bounds on the CTE (ii) role of the incentive μ in the demand for trust (iii) optimal number of nodes supported with incentives by a selfish/fair oracle and (iv) comparison of the demand for trust in various environments (equilibrium, incentives, non-equilibrium).

Collective Trust Elasticity of the trust demand function

We study the CTE of the trust demand function in three scenarios: (i) $\omega_1 > \omega_2$, where collective trust c_{coll} plays a greater role in the determination of the demand function for trust rather than oracle

selfishness θ (ii) $\omega_1 = \omega_2$, where collective trust c_{coll} is weighted equally alongside oracle selfishness θ in the determination of the demand for trust and (iii) $\omega_1 < \omega_2$, where collective trust c_{coll} plays a lesser role in the determination of the demand function for trust rather than oracle selfishness θ. In each of these scenarios, we study the elasticity of the collective trust c_{coll} with varying degrees of oracle selfishness θ. The level of selfishness is varied from $\theta = 0.1$ (fair), $\theta = 0.5$ (reasonable), $\theta = 0.8$ (selfish).

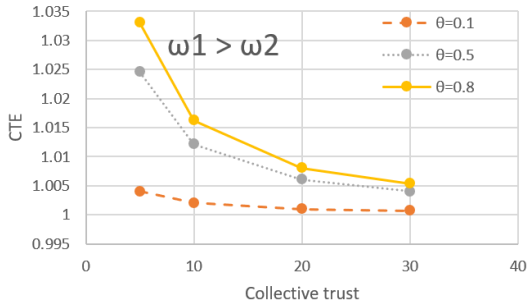

Figure 2. CTE, for $c_{coll} \gg$ selfishness θ

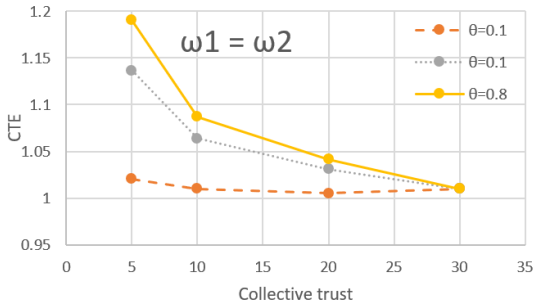

Figure 3. CTE, for $c_{coll} \sim$ selfishness θ.

(i) $\omega_1 > \omega_2$: Here, the collective trust c_{coll} plays a greater role in the determination of the demand function for trust than the oracle selfishness θ. We see that the CTE is lowest when the oracle is fair ($\theta = 0.1$), demonstrating almost unit elastic behavior. This indicates the amount of change in collective trust impacts the CTE by a similar amount. However, as θ increases, i.e. for an oracle that displays increasingly selfish tendencies, the CTE is higher denoting a highly elastic function. Thus, at smaller values of the collective trust (c_{coll}), the change in the amount of trust demanded from the oracle is high. Further, as the value of collective trust in the oracle increases, the CTE reduces indicating a smaller change in the amount of trust demanded from the oracle.

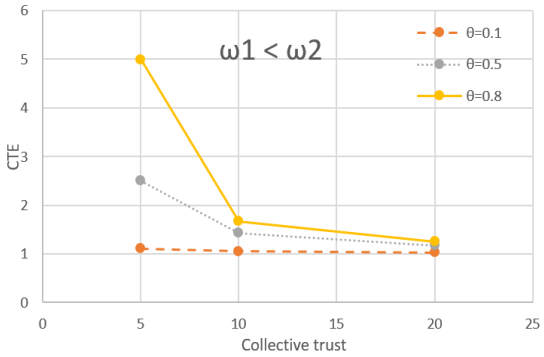

Figure 4. CTE, for $c_{coll} \gg$ selfishness θ

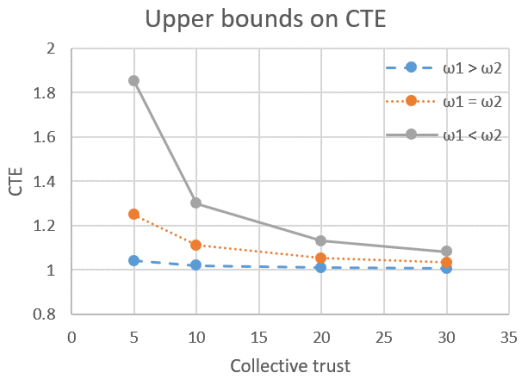

Figure 5. Upper bounds on the CTE

(ii) $\omega_1 = \omega_2$: When $\omega_1 = \omega_2$, the selfishness of the oracle and the collective trust play equal roles in the determination of trust placed in the oracle. Thus, the oracle is more selfish than when $\omega_1 > \omega_2$. We see the same general trends apply for the slope of the CTE, whereas collective trust increases, the change in the amount of trust demanded from the oracle is small. However, we see that for selfish oracle ($\theta = 0.5, 0.8$), the value of the CTE is higher than the previous case where $\omega_1 > \omega_2$. This finding is indicative of the increased selfishness of the oracle causing a general decrease in demand for trust.

(iii) $\omega_1 > \omega_2$: In this case, the oracle's selfishness plays a greater role in the determination of trust than the collective trust placed in the oracle. We see that, of the three cases ($\omega_1 > \omega_2, \omega_1 = \omega_2, \omega_1 < \omega_2$), the CTE is the highest indicating a large change in the amount of trust demanded by the nodes. However, the slope of the CTE function for $\theta = 0.1$ stays the same through all three cases, showing the optimal behavior of the system to be unit elastic for small values of oracle selfishness θ.

Figure 5 shows the bounds on the CTE as derived in equation (15). We see that the lower bound of the CTE is 1, indicating that the CTE in our model cannot be inelastic. An inelastic trust function implies that as the collective trust increases, the demand for the oracle's services will decrease. We study the variation of the upper bound of the CTE in the same three scenarios ($\omega_1 > \omega_2, \omega_1 = \omega_2, \omega_1 < \omega_2$). From Figure

4, we see that the CTE is the highest when collective trust is the lowest, and is consistent with the findings in Figures 2, 3, and 4.

Role of incentives in the demand for trust

Figure 6 shows the distribution of incentives μ as a function of the collective trust c_{coll}. We study the distribution of incentives μ under the same three scenarios as above (i) $\omega_1 > \omega_2$, (ii) $\omega_1 = \omega_2$, and (iii) $\omega_1 < \omega_2$.. We see that when $\omega_1 > \omega_2$, as the collective trust c_{coll} increases, the slope of the incentive function is close to zero. In contrast, the slope of the incentive function as a function of change in collective trust is largest for the case when $\omega_1 < \omega_2$. This shows that as the collective trust increases, the incentive is highest for the nodes to trust the oracle since the node's peers approve of the trustworthiness of the oracle. This finding is affirmed by the curve for $\omega_1 > \omega_2$, where the collective trust c_{coll} plays a greater role in determination of the trust function than oracle incompetence.

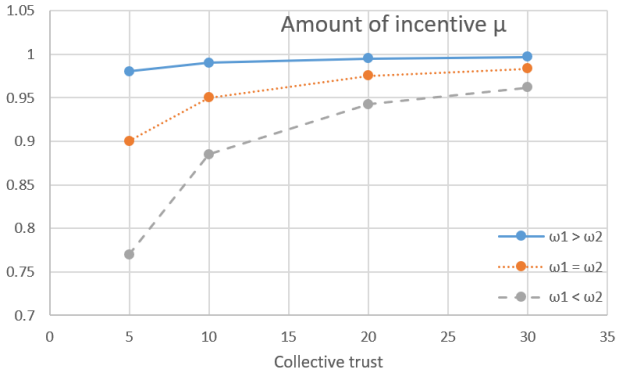

Figure 6. Incentive distribution as a function of c_{coll}. The higher the value of c_{coll}, higher is the incentive to trust the oracle.

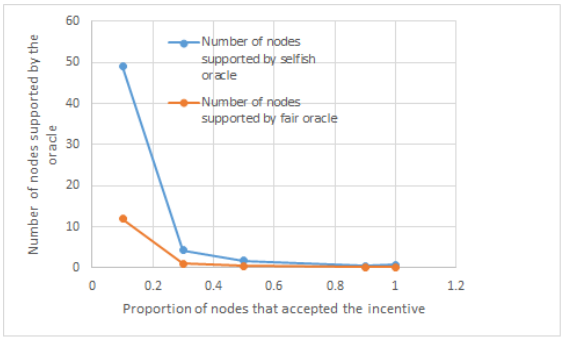

Figure 7. Comparison of selfish and fair modes of operation of the oracle with incentives for nodes. The oracle can support four times as many nodes in selfish mode as in fair mode.

Optimal number of nodes supported with incentives by a selfish/fair oracle

Figure 7 shows the performance of the system at equilibrium, where incentives based on the CTE are provided to ρ% of the nodes. The number of nodes supported by the oracle in both selfish and fair modes decreases exponentially as ρ increases. In particular, the decline in the number of nodes that can be

supported drops steeply when 30% or more of the nodes in the system accepted incentives. Thus, from Figure 7, for both the fair and selfish modes, the optimal value of ρ for highest number of nodes supported by an oracle is 10%. This shows that significantly better network design can be obtained by providing incentives to a smaller subset of the nodes, rather than relying on the entire system to respond to incentives. This smaller subset could be higher-performing nodes that achieve efficient smart contract utility. This finding is in line with work reported in Banerjee et al (2011) and Gneezy et al (2011), where it has that increasing taxes does not cause people to work less, or that salary caps do not cause top athletes and CEOs to put in lower-quality work, or that welfare benefits do not cause low-income individuals to stop working or looking for paid employment. Instead, people's motivations for working hard are shown to be related to more visceral motives – social status, dignity and reputation. While it is hard to quantify what the motives are for oracles or blockchain nodes to perform fairly, it is safe to say that offering incentives to the entire set of nodes and oracles might not be effective after all. Instead, offering incentives to a smaller subset who actually value it – whether it be computational resources or reputation-boosting reviews – may be more valuable than offering it to the entire network. For example, high-tier oracles may value both reputation and computational resources, and nodes may value additional computational resources than reputation.

Further, as expected, there is a striking difference between the number of nodes supported by the oracle in its fair and selfish modes of operation. Specifically, we see for low value of ρ $(< 50\%)$, in the selfish mode of operation, the oracle can support four times as many nodes as in its fair mode of operation. The reason is that in the selfish mode, the oracle can choose to exhaust fewer resources to satisfy the smart contract requests of the nodes. This selfishness is characteristic of a *self-preserving* oracle, thus causing the oracle to achieve more with less. In contrast, the fair mode of operation requires the oracle to be *self-effacing*, where it exhausts all its resources to satisfy the smart contract demands of its nodes. The results from Figure 7 demonstrate the "strain" placed on the oracle in the fair mode versus the selfish mode. Finally, we see that when 50% or more of the nodes in the system accept incentives, the performance of

the oracle in selfish and fair modes starts converging. For these levels of incentivized nodes, the oracle has to meet the collective trust requirements of the smart contracts generated by larger number of nodes, thereby causing it to serve fewer nodes in the system.

Comparison of the demand for trust in various environments

Figures 6 and 7 shows the results of simulating the demand function for varying number of nodes. The demand function in each chart is plotted in three modes: (i) Demand as a function of the number of the nodes in non-equilibrium with the oracle supply function (ii) Demand as a function of the number of the nodes in equilibrium with the oracle supply function for both selfish and fair oracles and (iii) Demand function with incentives. In keeping with the results obtained in Figure (7) above, which shows the optimal percentage of nodes receiving incentives is 10%, we set the value of ρ to be 0.1 for this simulation. Across Figures 8 and 9, we see that the demand function for trust from the oracle decreases as the oracle's selfishness increases. At equilibrium, when the supply of trust at the oracle matches the demand for trust from the nodes, the demand function is higher. The highest demand for trust from the oracle is obtained when nodes are offered incentives to trust the oracle.

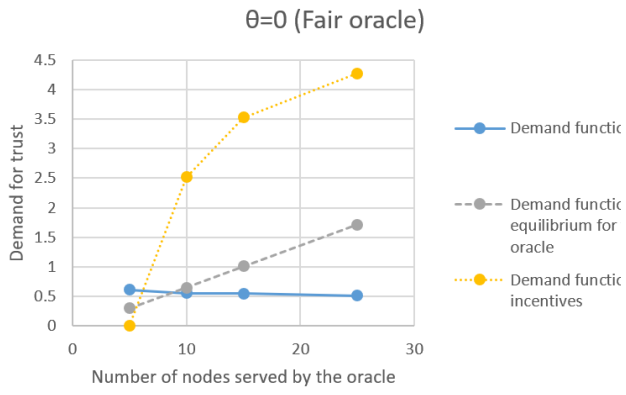

Figure 8. Demand for trust from fair oracle

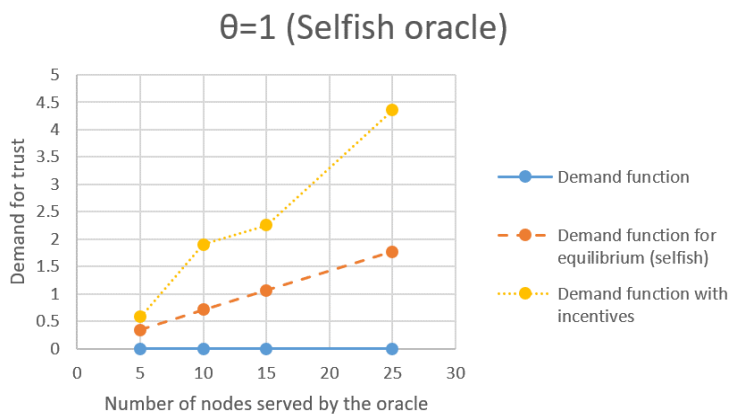

Figure 9. Demand for trust from selfish oracle

Discussion

Our model aims to understand trust in a blockchain system where nodes in a trusted environment have to rely on an oracle to obtain data from an untrusted environment. More broadly, our work relates to the nature of trust in digital economies. By commoditizing trust, we created a model where trust can be traded like a commodity. Further, the behavior of the system can be enhanced by offering incentives to the nodes and the oracle to choose fairness in service, akin to a service-level agreement for the oracle. In order to better understand the implications of this model, it is useful to take a deeper look at the structure of our model. First, we tackle the notion of the collective trust elasticity (CTE). Our model studies the sensitivity of the demand function to the collective trust as a means of determining the impact of trust on the demand for the oracle's services. However, other parameters in the demand functions such as number of nodes n, or the oracle's selfishness may be alternative or complementary metrics to determine the sensitivity of the demand function.

Second, we use the bounds of the CTE to determine incentives. Specifically, we chose a mean value of oracle selfishness ($\theta = 0.5$) to obtain the incentive μ, as a way of assuring the nodes that the oracle would choose to honor an average level of fairness and reduce extreme selfishness. There are a number of reasons why the selfishness value could be set. For example, nodes might set a certain value of selfishness as a precursor to entering a smart contract. Alternately, the oracle might be able to vary the selfishness based on factors such as priority, scheduling policies and load balancing. Further based on these criteria, the oracle may also choose to be selectively selfish, such as being fairer to some nodes more than others.

Third, our simulations showed that incentives are only useful when offered to a smaller subset of the population. Criteria for choosing this smaller subset of nodes might include load balancing among nodes, computational complexity of fetching data at the oracle, and security. Additionally, the cooperative work between the oracle and the nodes might be used to introduce social incentives such as reputation, that can track node and oracle behavior over time and provide a ratings-based system for better performance.

Finally, our results showed that oracles can support more nodes when they are selfish and fewer nodes when they are fair. In particular, the number of nodes supported by a selfish oracle is four times as much

as by a fair node. However, there might be applications where the number of nodes supported can be further increased by introducing delegation. Tiered, clustered or hierarchical topologies may reduce the burden of the oracle by delegating the intermediate smart contract generation and maintenance activities to cluster-heads at lower levels. Such distributed management of the oracle-node smart contract communication tasks can enhance the performance of the oracle by freeing up resources for efficient operations.

While this work has focused exclusively for oracles in blockchain environments, the implications of this work can be readily extended to networks and systems where a central entity delivers services for the network. Examples of such applications include trusted third parties in distributed systems such as cyberphysical systems and IoT networks (Eom et al. 2020) and (You et al., 2018), cluster-heads in wireless sensor networks and robotic applications with heterogeneous nodes where some nodes are responsible for higher-level tasks such as aggregation and data analysis, and in real-time distributed systems (Mbarek et al. 2016). In these applications, the affordances of our model concerning dynamically evolving trust, the option to act independently of peer influence and the role of incentives can all be leveraged to create robust environments that can function in the presence of malicious central points of failure. The notion of trust is a cross-disciplinary construct, and as such, our model for commoditization of trust can be used in multi-agent scenarios comprising of agents with mixed levels of resources.

CONCLUSION

The work in this paper concerns itself with the case of a single oracle serving a blockchain according to a commoditized trust mechanism that considers the selfishness of the oracle, and the collective trust placed by the blockchain in the oracle. This framework is scalable to the level of individual trust preferences that affect the collective trust, and the oracle has the ability to tune its selfishness to reflect its resource-conserving strategies. However, this framework has several limitations, as we outline below:

Multiple oracles: The applications of distributed ledger computing in various domains necessitate cross-domain trust and data processing. Oracles for different applications might be collocated for fetching data from various untrusted domains. Further, the computational resources of these oracles, the size of the network, load distribution and their selfishness metrics might vary. This creates a market for oracles of varying capabilities and blockchains for various applications to position themselves in market for trust. Thus, multiple directions for future work are envisioned. Oracles might adapt to a self-regulating market, or additional transaction models involving auctions of oracles' services might be envisioned. Additionally, improved contracts fashioned for inter-oracle communication might involve factors such as congestion control and multiplexing to allow for improved trust supplied by the oracles.

Hierarchy of oracles: While our framework involved a single oracle, a hierarchy of oracles organized in various tiers according to the level of trust required for the application might be envisioned for applications with differing QoS criteria. Such an architecture might have applications with highest trust delivery will have in the root of a hierarchy, with branches for applications with lower trust delivery. Alternately, the applications with similar trust requirements could be classified and processed according to a clustering mechanism or placed in different queues with differing time quantum per queue allocated according to the trust assigned for the queue.

Security: Our framework allows for an oracle to exhibit selfishness so that it may choose to preserve its resources. However, additional work is required to study the security implications of an oracle that goes rogue and compromises the security of the network. Additional work in the direction of malicious nodes will help illuminate the security implications of having an oracle in the trusted position of data purveyor from an untrusted environment.

In this paper, we proposed a model for studying the interplay of an oracle with a trusted blockchain and untrusted external environment. Our model borrowed upon constructs from social models of trust to study the role of selfishness and incentives in creating a dynamic framework where the blockchain can vary the trust placed in the oracle. The work in this paper provides multiple avenues for exploring new research

areas in smart contracts that are created with entities such as oracles that serve as intermediaries between two environments with differing levels of trust. This includes fundamental questions such as network topology, security of data, malicious nodes and oracles and design of incentives. As distributed ledger applications continue to rise in popularity and acceptance, other potential areas of research include the design of machine learning algorithms for generating smart contracts that utilize the services of oracles in differing ledger platforms and applications.

REFERENCES

Adler, J., Berryhill, R., Veneris, A., Poulos, Z., Veira, N., & Kastania, A. (2018). Astraea: A decentralized blockchain oracle. *arXiv preprint arXiv:1808.00528*.

Anderson, P. L., McLellan, R. D., Overton, J. P., & Wolfram, G. L. (1997). Price elasticity of demand. *McKinac Center for Public Policy. Accessed October*, *13*, 2010.

Allen, D. W., Lane, A., & Poblet, M. (2019). The governance of blockchain dispute resolution. *Available at SSRN*.

Banerjee, A. V., Banerjee, A., & Duflo, E. (2011). *Poor economics: A radical rethinking of the way to fight global poverty*. Public Affairs.

Barber, Bernard. a:1980. Informed Consent. Rutgers University Press.

Beck, R., Stenum Czepluch, J., Lollike, N., & Malone, S. (2016). Blockchain–the gateway to trust-free cryptographic transactions.

Bénabou, R., & Tirole, J. (2006). Incentives and prosocial behavior. *American economic review*, *96*(5), 1652-1678.

Bhattacharya, R., Devinney, T. M., & Pillutla, M. M. (1998). A formal model of trust based on outcomes. *Academy of management review*, *23*(3), 459-472.

Bowles, S., & Polania-Reyes, S. (2012). Economic incentives and social preferences: substitutes or complements? *Journal of Economic Literature*, *50*(2), 368-425.

Chatterjee, K., Goharshady, A. K., & Pourdamghani, A. (2019). Probabilistic smart contracts: Secure randomness on the blockchain. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)* (pp. 403-412).

Cooper, J. C. (2003). Price elasticity of demand for crude oil: estimates for 23 countries. *OPEC review*, *27*(1), 1-8.

Czap, H. (2005). Agent based computational model of trust. *Self-Organization and Autonomic Informatics (I)*, *135*, 160.

de Pedro, A. S., Levi, D., & Cuende, L. I. (2017). Witnet: A Decentralized Oracle Network Protocol. *arXiv preprint arXiv:1711.09756*.

Ding, L., Kolari, P., Ganjugunte, S., XXXXX, T., & Joshi, A. (2004, July). Modeling and evaluating trust network inference. In *Seventh International Workshop on Trust in Agent Societies at AAMAS 2004*.

Eom, S., & Lee, K. H. (2020). Incorporating Spatial Queries into Semantic Sensor Streams on the Internet of Things. In *Securing the Internet of Things: Concepts, Methodologies, Tools, and Applications* (pp. 1703-1719). IGI Global.

Freeman, R., Lahaie, S., & Pennock, D. M. (2017, February). Crowdsourced outcome determination in prediction markets. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Gächter, S., Nosenzo, D., & Sefton, M. (2013). Peer effects in pro-social behavior: Social norms or social preferences? *Journal of the European Economic Association*, *11*(3), 548-573.

Gneezy, U., Meier, S., & Rey-Biel, P. (2011). When and why incentives (don't) work to modify behavior. *Journal of Economic Perspectives*, *25*(4), 191-210.

Goel, N., Filos-Ratsikas, A., & Faltings, B. (2019). Decentralized Oracles via Peer-Prediction in the Presence of Lying Incentives.

Gupta, P., & Kumar, P. R. (2000). The capacity of wireless networks. IEEE Transactions on information theory, 46(2), 388-404.

Hawlitschek, F., Teubner, T., & Weinhardt, C. (2016). Trust in the sharing economy. *Die Unternehmung*, *70*(1), 26-44.

Idelberger, F., Governatori, G., Riveret, R., & Sartor, G. (2016, July). Evaluation of logic-based smart contracts for blockchain systems. In *International Symposium on Rules and Rule Markup Languages for the Semantic Web* (pp. 167-183).

Kirschen, D. S., Strbac, G., Cumperayot, P., & de Paiva Mendes, D. (2000). Factoring the elasticity of demand in electricity prices. *IEEE Transactions on Power Systems*, *15*(2), 612-617.

Koszegi, B. (2014). Behavioral contract theory. *Journal of Economic Literature*, *52*(4), 1075-1118.

Lewis, J. D., & Weigert, A. (1985). Trust as a social reality. *Social forces*, *63*(4), 967-985.

Luhmann, N. 1979. Trust and Power. Wiley.

Khambatti, M., Dasgupta, P., & Ryu, K. D. (2004, April). A role-based trust model for peer-to-peer communities and dynamic coalitions. In *Second IEEE International Information Assurance Workshop, 2004. Proceedings.* (pp. 141-154).

Kovářík, J., Brañas-Garza, P., Cobo-Reyes, R., Espinosa, M. P., Jiménez, N., & Ponti, G. (2012). Prosocial norms and degree heterogeneity in social networks. *Physica A: Statistical mechanics and its Applications*, *391*(3), 849-853.

Krug, J., Zoltu, M., Williams, A. K., & Alexander, S. (2015). Augur: a decentralized oracle and prediction market platform. *arXiv preprint arXiv:1501.01042*.

Macy, M. W., & Skvoretz, J. (1998). The evolution of trust and cooperation between strangers: A computational model. *American Sociological Review*, 638-660.

Magazzeni, D., McBurney, P., & Nash, W. (2017). Validation and verification of smart contracts: A research agenda. *Computer*, *50*(9), 50-57.

M'barek, S., Baccouche, L., & Ben Ghezala, H. (2016). Model Driven Engineering for Quality of Service Management: A Research Note on the Case of Real-Time Database Management Systems. *Journal of Database Management*, *27*(4), 24-38.

Medo, M., Zhang, Y. C., & Zhou, T. (2009). Adaptive model for recommendation of news. *EPL Europhysics Letters*, *88*(3), 38005.

Mui, L., Mohtashemi, M., & Halberstadt, A. (2002, January). A computational model of trust and reputation. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences* (pp. 2431-2439).

Nah, F. F., Schiller, S. Z., Mennecke, B. E., Siau, K., Eschenbrenner, B., & Sattayanuwat, P. (2017). Collaboration in Virtual Worlds: Impact of Task Complexity on Team Trust and Satisfaction. *Journal of Database Management (JDM), 28*(4), 60-78. doi:10.4018/JDM.2017100104

Naoumov, V., & Gross, T. (2003). Simulation of large ad hoc networks. In Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems (pp. 50-57). ACM.

Nehai, Z., Piriou, P. Y., & Daumas, F. (2018, July). Model-checking of smart contracts. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 980-987). IEEE.

Nixon, P., Wagealla, W., English, C., & Terzis, S. (2005). Security, privacy, and trust issues in smart environments.

Packer, H. S., Drăgan, L., & Moreau, L. (2014). An auditable reputation service for collective adaptive systems. In *Social Collective Intelligence* (pp. 159-184). Springer.

Parsons, T. a:1963. On the Concept of Power. Proceedings of the American Philosophical Society 107:232-62.

Peterson, J., Krug, J., Zoltu, M., Williams, A. K., & Alexander, S. (2015). Augur: a decentralized oracle and prediction market platform. *arXiv preprint arXiv:1501.01042*.

Rajan, U., Seru, A., & Vig, V. (2010). Statistical default models and incentives. *American Economic Review*, *100*(2), 506-10.

Rassenfosse, G. D., & Potterie, B. V. P. D. L. (2012). On the price elasticity of demand for patents. *Oxford Bulletin of Economics and Statistics*, *74*(1), 58-77.

Scherer, M. (2017). Performance and scalability of blockchain networks and smart contracts.

Sheth, A., & Subramanian, H. (2019). Blockchain and contract theory: modeling smart contracts using insurance markets. *Managerial Finance*.

Simmel, G. a:1900. The Philosophy of Money. Routledge & Kegan Paul, 1978.

Siau, K., Long, Y., & Ling, M. (2010). Toward a Unified Model of Information Systems Development Success. *Journal of Database Management (JDM), 21*(1), 80-101. doi:10.4018/jdm.2010112304.

Sliwka, D. (2007). Trust as a signal of a social norm and the hidden costs of incentive schemes. *American Economic Review*, *97*(3), 999-1012.

Subramanian, H. (2018). Decentralized blockchain-based electronic marketplaces. *Commun. ACM*, *61*(1), 78-84.

Sutcliffe, A., & Wang, D. (2012). Computational modelling of trust and social relationships. *Journal of Artificial Societies and Social Simulation*, *15*(1), 3.

Tian, D., & Georganas, N. D. (2002, September). A coverage-preserving node scheduling scheme for large wireless sensor networks. In Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (pp. 32-41). ACM.

Tsankov, P., Dan, A., Drachsler-Cohen, D., Gervais, A., Buenzli, F., & Vechev, M. (2018, January). Securify: Practical security analysis of smart contracts. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (pp. 67-82).

Venanzi, M., Rogers, A., & Jennings, N. R. (2013, May). Trust-based fusion of untrustworthy information in crowdsourcing applications. In Proceedings of the *International Conference on Autonomous Agents and Multi-agent Systems* (pp. 829-836).

Wang, K., Qi, X., Shu, L., Deng, D. J., & Rodrigues, J. J. (2016). Toward trustworthy crowdsourcing in the social internet of things. *IEEE Wireless Communications*, *23*(5), 30-36.

Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., & Wang, F. Y. (2019). Blockchain-enabled smart contracts: architecture, applications, and future trends. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *49*(11), 2266-2277.

Wang, S., Yuan, Y., Wang, X., Li, J., Qin, R., & Wang, F. Y. (2018, June). An overview of smart contracts: architecture, applications, and future trends. In *2018 IEEE Intelligent Vehicles Symposium (IV)* (pp. 108-113).

Watanabe, H., Fujimura, S., Nakadaira, A., Miyazaki, Y., Akutsu, A., & Kishigami, J. (2016, January). Blockchain contract: Securing a blockchain applied to smart contracts. In Proceedings of the *IEEE International Conference on Consumer Electronics (ICCE)* (pp. 467-468).

Weaver, A. C., Boyle, J. P., & Besaleva, L. I. (2012, July). Applications and trust issues when crowdsourcing a crisis. In Proceedings of the *21st International Conference on Computer Communications and Networks (ICCCN)* (pp. 1-5).

Wöhrer, M., & Zdun, U. (2018). Design patterns for smart contracts in the Ethereum ecosystem.

Ye, H. J., & Kankanhalli, A. (2017). Solvers' participation in crowdsourcing platforms: Examining the impacts of trust, and benefit and cost factors. *The Journal of Strategic Information Systems*, *26*(2), 101-117.

You, X., Li, Y., Zhu, Z., Yu, L., & Sun, D. (2018). QGLG Automatic Energy Gear-Shifting Mechanism with Flexible QoS Constraint in Cyber-Physical Systems: Designing, Analysis, and Evaluation. *Journal of Database Management (JDM)*, *29*(1), 43-65.

Yu, H., Shen, Z., Miao, C., & An, B. (2012, December). Challenges and opportunities for trust management in crowdsourcing. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 02* (pp. 486-493).

Zou, J., Ye, B., Qu, L., Wang, Y., Orgun, M. A., & Li, L. (2018). A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services. *IEEE Transactions on Services Computing*.

Zhang, F., Cecchetti, E., Croman, K., Juels, A., & Shi, E. (2016, October). Town crier: An authenticated data feed for smart contracts. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 270-282.