CryptoTask - Decentralized Task Market

Vedran Kajić, Ivan Nanut

Table of Contents

1 Abstract

We propose a decentralized task market based on blockchain technology that disrupts current freelancing systems. The basis is a consensus mechanism similar to that used in existing prediction market projects such as Augur and Gnosis, with some important differences and innovations that enable scalability, an issue plaguing most of the popular blockchain projects. One key innovation is a reviewer board selection mechanism built into the blockchain, including an escalation option that satisfies Nash equilibrium. The selection mechanism satisfies the condition of secrecy in order to discourage potential collusion and off-chain communication between reviewers by implementing a reporting mechanism. Voting is done in two stages, including a secret commit phase, to dissuade parties from using the waiting strategy.

2 Introduction

2.1 Blockchain

Bitcoin [1] appeared in 2009 and started the blockchain revolution, decentralizing basic finance. Soon, other projects followed, with the main idea of decentralizing more than just the monetary system. That culminated with Ethereum [2] which is a decentralized Turing complete machine, that theoretically allows other projects to implement their logic on the Ethereum platform, instead of building a new blockchain from scratch. However, scalability remains the main issue.

2.2 Untapped freelancing market

We propose applying the blockchain technology to the current freelancing market. The commercial potential is huge, with freelancers making up 35% of the workforce in the U.S. and contributing approximately 1 trillion USD to the economy [3]. The decentralization that blockchain enables offers multiple advantages: lower fees, no censorship, dispute process transparency, no financial limits, no arbitrary meddling from corporations or governments.

2.3 Comparison to prediction markets and other relevant projects

Prediction market projects based on blockchain are the most relevant to the proposed system. These markets are created for the purpose of trading the outcome of events and can be used as oracles as well. The reason for relevancy is that prediction markets' event decision process is based on a consensus mechanism similar to the one proposed here. It is, however, easier to design the proposed system robustly than a general prediction market, as the value of each task is defined in advance, unlike with prediction markets where the value of the outcome changes over time as traders place bets. Also, on task vs event basis, task markets scale more easily as review processes are triggered only in case of disputes; that is to say for most cases the task creator and freelancer will agree on the task result and payment without the need for arbitration, while prediction markets always require an oracle. Additionally, unlike prediction markets, parasite attacks are not a concern, as the system serves the purpose of settling potential party disputes, so there is no public result worth stealing.
For example, a potential Truthcoin [4] market attack would require a significant stake of voting coin, however the attack revenue is nearly unlimited. There is also a parameter tau, related to the price of decision-slots, complicating resolution of events in the far future. Overall, the system is complicated, and there is currently no implementation.
Augur [5] prediction market does not have a dispute mechanism well-defined in the original white paper, however recent blog posts [6] suggest a hierarchical escalation mechanism similar to the one we

propose. It is unclear how the reporting positions are filled and whether it is known in advance who the reporters are. That is important because if reporters know who the other reporters are, they can communicate prior to voting. From a game theory perspective it could be argued that this is irrelevant as the escalation mechanism ensures that any collusion would be irrational. However, especially in the task market case, there are reasons why it is better to avoid the possibility of reviewer communication, such as: tasks completed, but quality is poor, reviewers higher up are not as skilled in the task domain as specialized reviewers are, etc.
Gnosis [7] is another Ethereum-based prediction market, that is likely to use the “Ultimate oracle” scheme. The idea is that a fixed bet is placed on an outcome (say 100 ETH), and then any holder can challenge the decision by placing a larger bet. We propose using a similar system for dispute resolution, however, only as the final step in the hierarchical dispute resolution, and not with a fixed initial deposit, but related to the disputed task value. While, theoretically, the ultimate oracle could be the only dispute mechanism, there are advantages to having first randomly selected reviewer panels, such as: better understanding of the task domain, no time bias (reviewers can't see what others voted for) and additionally, any larger amount for the initial deposit might be out of reach for the ordinary system users. On the other hand, reviewers are professionals holding reasonable coin stake, and they are in a better position to escalate the dispute than the average freelancer. The reasons for having a dispute hierarchy are similar to those for having a hierarchy in classical legal systems, and why there is not only the supreme court.
Stox [8] is yet another proposed Ethereum-based prediction market, with dispute architecture not yet defined.
Blocklancer [9] is the only blockchain task/freelancing market project that we are aware of. The dispute mechanism is basic, requiring all stake holders to vote on all disputes, thus it is difficult to achieve scalability.

Though some principles are shared in the aforementioned projects and the proposed system, none provide a satisfying solution for our goal. Thus, we decided to design a system that will implement all required mechanisms efficiently, while providing greater scalability.

3 Technical

3.1 Overview

We introduce a blockchain-based task market system consisting of clients, freelancers and reviewers. Clients post job offers, freelancers apply for these and reviewers are stake-holders that have a chance of being selected into a review panel proportional to their stake, thus preventing sibyl attacks; in the case of a task dispute for which they were selected into the review panel, they are required to cast a vote on whether the task was completed or not. Reviewers will want to define their areas of expertise honestly, otherwise they risk losing in the consensus mechanism. In the simplest case, the client and the freelancer will agree on the job outcome. If the job is completed, the freelancer gets paid, and if it isn't completed or the deadline expired, the client gets a refund and the freelancer's deposit. Clients put a deposit equal to the task value plus 10% to cover the potential dispute costs if they turn out to be the losing party, while freelancers need to put down 10% of the task they applied for. The freelancer deposit is used as a measure against wasting the client's time, as without it, freelancers could DOS the system. It is also used to initiate disputes, as it will be paid to reviewers as compensation. In case of a dispute, depending on the result, either the freelancer or the client will lose 10% of the task value, that will go to the reviewers.

Review panels are created for each dispute and consist of randomly selected reviewers. There are 10 reviewers on average in the panel, though that number is not deterministic. Reviewers are not aware of other reviewers. Voting is done in two stages: commit and reveal; the idea being that no reviewer can wait to see how other reviewers voted. The proposed system satisfies Nash equilibrium as no party has incentive to change their game strategy, considering that other parties keep their strategies fixed.

The platform is already deployed on Ethereum main-net. Long term, Monero hard fork is in works for strongly privacy oriented users, as ring signatures hide the sending address, making the task creator unknown, as well as reviewers when applying for the panel after learning that they were selected. The reviewer selection mechanism is built into the blockchain; it could be thought of as a hard-coded smart contract.

The system does not rely on reputation to assure fair task evaluation. While reputation can be used as part of the task applicant selection process on the task creator's part, it is not a necessary component that ensures system soundness. Systems relying on reputation, among other complexities, such as potential Sybil attacks, face the problem of initial reputation. That is, a system can be unstable if there are no members with established reputation. On the other hand, giving some initial reputation to selected users can be arbitrary and goes against the decentralized philosophy.

In *Figure 1* the system overview is given. There is a number of task domains, and each reviewer can apply for one or more domains. In case that the freelancer and the client cannot agree on the task submission, a dispute process can be initiated by either party. A review panel is formed consisting of 10 reviewers on average, *R1* to *R10*. *s_ri* is the *ith* reviewer's private key. *hash(n+d)* is *(n+d)_block*'s hash. Each reviewer computes a value on *(n+d)_block*'s hash with their private key (signing): RSA(*hash(n+d)*, *s_ri*).
A certain number of starting bits defines the task number (in case that more than one dispute was initiated in that block), while the ending bits are used to determine whether the reviewer was selected. The likelihood of being selected is proportional to the reviewer's stake. It is important to note that at this stage only each reviewer knows this information for themselves, but can prove it to everyone later by revealing the obtained value that anyone can later use with that reviewer's public key to check that the final value is equal to the *(n+d)_block*'s hash. After the review panel has reached its decision, either the client, the freelancer or any member of the review panel can dispute that decision and escalate to the “Ultimate oracle” consisting of all the stakeholders.

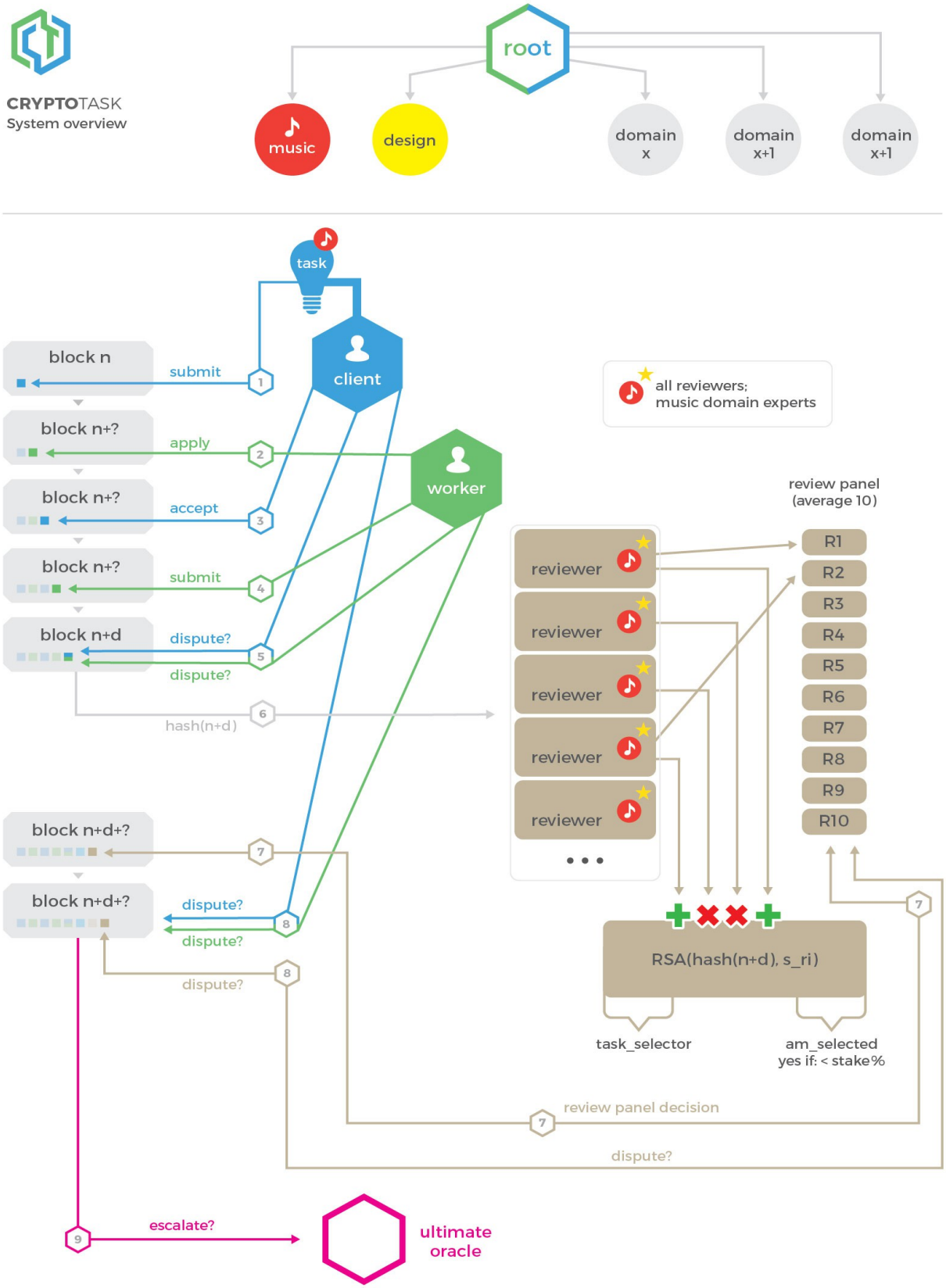

Figure 1 System overview

3.2 Dispute process and reviewer panel selection

As the dispute mechanism is a critical system component, we will go into greater detail.

At each block, each reviewer encrypts the last block hash with his private key, and if the last part is smaller than his reviewer stake percentage, he knows that he is selected. The main idea here is that only he knows that fact. That can later be proven to anyone without revealing the private key, simply by giving the encrypted value, while anyone can encrypt it with the matching public key, thus checking that the obtained number matches the original number, that is the original block hash. The reviewer then proceeds to submit the vote hash, and then in the reveal phase he proves that he was selected and reveals the vote. If there are more disputes, the starting part of the encrypted last block's hash can be divided into task ranges.

Any party that tries to communicate off-chain can be punished by anyone using the breach mechanism, resulting in the offending party's loss of deposit. For example, any attempt by a selected reviewer to provide proof-of-selection as the first step towards collusion can be reported as breach.

The result given by the review panel can be challenged by either the freelancer, the client or one of the reviewers with the deposit equaling the task value. The higher instance can either be another review panel or a mechanism similar to Gnosis “Ultimate oracle” which includes all the coin stake holders.

The task solution is submitted publicly, but the encryption mechanism can be added so that only the client can see the solution and is revealed only in case of a dispute. The client generates a one time public-private key pair and announces his one time public key that the freelancer uses to encrypt the solution. In case of a dispute, the client reveals his one time private key.

3.3 Potential attacks

Let us consider a dispute and the probability of an attacker getting more than 50% of review panel slots, depending on the total review stake he holds.
n is the number of reviewer positions the attacker holds, m is the number of reviewer positions honest reviewers hold (total - n), while p is 10 divided by the total number of reviewers, as the system selects a review panel with 10 members on average.
The probability of an attacker getting i slots is then given by the binomial distribution:

$$P_A(i) = \binom{n}{i} p^i (1-p)^i \tag{1}$$

The probability of honest reviewers getting j slots is:

$$P_H(j) = \binom{m}{j} p^j (1-p)^j \tag{2}$$

We are interested in finding the cumulative distribution of the difference of these two binomial distributions for positive values, i.e. when the attacker controls more slots than honest reviewers [10]:

$$P_D = \sum_{i=0}^{n} (1-p)^{n-i} p^i (1-p)^m \binom{n}{i} \sum_{j=0}^{m} (-1)^j \binom{m}{j} \frac{(-n+i)_j}{(1+i)_j} \left(\frac{p^2}{(-1+p)^2}\right)^j \tag{3}$$

That expression gives us the probability of an attacker getting more than 50% of review panel slots. We can generate a table with these probabilities for various attack review stakes. For example, if there are 1000 reviewers and the attacker controls 50 (5% stake), then n is 50, m is 950 and p is 0.01.

Review stake %	Probability of >50% in a dispute review panel (P_D)
5%	0.015%
10%	0.12%
15%	0.48%
20%	1.43%
25%	3.46%
30%	7.13%
35%	12.93%
40%	21.13%
45%	31.54%

As we can see from the table, even for a significant attack stake, the probability of getting more than 50% of review slots is very low.

Now let us consider an attack where the attacker controls a significant percentage of mining power (p_A) as well, and can thus decide whether to broadcast a mined block.
The idea behind a combined attack is that as the review panel selection is done pseudo-randomly based on a block hash, whoever mined the block can know how many review panel slots he was given (in case he holds review stake, of course). They have no way of knowing how many review slots were given to other reviewers. If we ignore the block reward and mining fees lost, the attacker will try to find an optimal "review slots obtained" threshold under which he will not broadcast mined blocks. So if they mined a block, there are three possibilities: they will broadcast the block ($P_s P_{Ds}$), not broadcast it and risk that honest miners will mine the next block ($p_H(1-P_s)P_D$) or they will mine the next block again ($p_A(1-P_s)$), in which case everything repeats, and so ad infinitum.
If honest miners control p_H mining share, the probability of a combined stake and mining attack giving more than 50% review panel slots to the attacker is:

$$P_C = p_H P_D + p_A(P_s P_{Ds} + p_H(1-P_s)P_D + p_A(1-P_s)(P_s P_{Ds} + \ldots)) \tag{4}$$

P_s is cumulative binomial distribution that gives the probability of getting at least *s* slots ("review slots obtained" threshold) in the review panel:

$$P_s = \sum_{i=s}^{n} \binom{n}{i} p^i (1-p)^i \tag{5}$$

P_{Ds} is the cumulative distribution of the difference of two binomial distributions, but with the attacker binomial distribution clipped for values lower than *s*. The attacker would choose an optimal *s*

that maximizes the attack probability P_C. Finding that *s* is difficult as we cannot derive (4) as *s* appears as summation bound. What we can do is show the upper limit on the attack probability. Let us assume the worst case, that the attacker controls 50% of the total mining power, thus $p_A = p_H = 0.5$. $P_s P_{Ds} \leq P_D$ holds as $P_s P_{Ds}$ is a subset of all the cases when attacker can have majority (P_D). Obviously $(1 - P_s) P_D \leq P_D$ is true. Thus:

$$\begin{aligned} P_C \leq \frac{1}{2} P_D + \frac{1}{2}\left(P_D + \frac{1}{2} P_D + \frac{1}{2}\left(P_D + \frac{1}{2} P_D + \frac{1}{2}(\ldots)\right)\right) = \ldots \\ \ldots = \frac{1}{2} P_D + \frac{1}{2} P_D \left(1 + \frac{1}{2} + \frac{1}{2}\left(1 + \frac{1}{2} + \frac{1}{2}(\ldots)\right)\right) \end{aligned} \tag{6}$$

This infinite sum appears twice in (6):

$$\sum_{i=1}^{\infty} \frac{1}{2^i} = 1 \tag{7}$$

We can simplify (6) using (7) to:

$$P_C \leq 2 P_D \tag{8}$$

This means that even if the attacker controls 50% of the mining power, the combined attack chance does not increase more than twice compared to the pure stake attack.

A logical question is why even analyze attacks on the review panel stage of the dispute mechanism, as it is always possible to escalate to the “Ultimate oracle”, where all stakeholders can vote, and assuming at least 51% are honest, that should dissuade any attacker.
One of the reasons why it is better to catch as many as possible attack attempts, as well as honest disputes, in the first dispute phase is that there exist gray areas, meaning a task can be done, but of poor quality, and that, combined with the time bias of the “Ultimate oracle”, would mean that future reviews are influenced heavily by prior reviews. Reviewers can also afford to escalate the dispute, which can be out of reach for many freelancers. It is worth mentioning that adding a third state between task “done” and “not done”, for example “unclear”, does not solve the problem of gray areas. There simply exist two more gray areas; between “done” and “unclear”, and between “unclear” and “not done”.

4 Conclusion

We present a novel system that solves scalability issues present in current prediction markets and freelancing systems, letting us target the commercial freelancing economy, offering a number of advantages such as lower fees, no censorship and other blockchain related advantages. Considering the size of the freelancing economy, the commercial potential is huge.

5 References

1: Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008, https://bitcoin.org/bitcoin.pdf
2: Vitalik Buterin, A Next-Generation Smart Contract and Decentralized Application Platform, 2013, https://github.com/ethereum/wiki/wiki/White-Paper
3: Freelancers Union, New Study Finds Freelance Economy Grew to 55 Million Americans This Year, 35% of Total U.S. Workforce, 2016, http://www.marketwired.com/press-release/new-study-finds-freelance-economy-grew-55-million-americans-this-year-35-total-us-workforce-2164446.htm
4: Paul Sztorc, Truthcoin, , http://www.truthcoin.info/papers/truthcoin-whitepaper.pdf
5: Jack Peterson, Joseph Krug, Augur: a Decentralized, Open-Source Platform for Prediction Markets, , https://bravenewcoin.com/assets/Whitepapers/Augur-A-Decentralized-Open-Source-Platform-for-Prediction-Markets.pdf
6: , A Roadmap For Augur and What's Next, 2017, https://medium.com/@AugurProject/a-roadmap-for-augur-and-whats-next-930fe6c7f75a
7: , Gnosis, , https://gnosis.pm/resources/default/pdf/gnosis_whitepaper.pdf
8: , Stox Platform forPrediction Markets, , https://www.stox.com/assets/stox-whitepaper.pdf
9: , BlockLancer, , https://blocklancer.net/static/main/docs/lancer_whitepaper.pdf
10: , Difference of two binomial random variables, , https://math.stackexchange.com/questions/562119/difference-of-two-binomial-random-variables