Decentralized Arbitration Court White Paper v0.1

Lesaege Clément

April 2016

1 Introduction

Smart contracts are smart enough to execute as programmed. However they are not smart enough to depend on elements outside the blockchain or render subjective judgements.

Outside elements can be included inside the blockchain relying on oracles. Those can either be trusted third parties(3) or decentralized organizations relying on game theoretic incentives to have their members submit honest information. The later method is used by the prediction market projects Augur(8) and Gnosis(2). Building a smart contract allowing arbitration by a trusted third party is straightforward and can be done on Bitcoin(7) using multisignature addresses(1). The goal of the Decentralized Arbitration Court is to be a decentralized organisation providing arbitration services while relying on game theoretic incentives to have arbitrators and juries ruling cases correctly.

We cannot simply transpose previous models used for prediction markets because:

- A prediction market oracle output is generally relevant to many parties (**?**), while an arbitration result is generally relevant to few parties.
- A prediction market oracle output is public, while parties may want to limit the spread of dispute information to the relevant parties (arbitrators and juries).
- A prediction market oracle output tends to have a low level of subjectivity, while arbitration results can sometimes be highly subjective.
- In prediction markets, gathering information tends to be low cost and is a one time process, while in arbitration the cost is higher and involves question and response between the arbitrator and parties. This implies more scalability issues.

Arbitration courts involves new challenges and we aim to solve them in this white paper.

2 Previous Work

2.1 Arbitration through multi-signature addresses

2.2 Schelling-coin mechanism

(5) (9)

3 Project Description

3.1 Arbitrated contract

The Arbitration Court is an opt-in system. This means that smart contract developers must specify how the court can interact with them. They don't need to give them full control over their contracts. In a case of a simple buying contract involving two parties, the contract could allow the court to give the ether belonging to the contract either the buyer or the seller but not allow the court to choose another party to receive the funds.

3.1.1 Hiding plain English contract

In order for an arbitrator or a jury member to arbitrate a contract, he needs to access the plain English (or other language, see the part on subcourts) version of the contract. Stocking the plain English version of the contract on the blockchain would have an high gaz cost and would cause privacy issues allowing anyone to access it even when the contract does not involve a dispute.

In order to avoid this issue, we use the notion of commitment. A commitment allow parties to commit to a specific information without revealing it during the commitment phase. If needed, a party may reveal it later and prove that the revealed information correspond to the information committed. In our system, both parties will agree on a plain English version of the contract. The party creating the contract will input an hash of the plain English contract. The other party will be able to verify that the hash correspond to the plain English version of the contract before interacting with it. In case of dispute, each party will be able to reveal the plain English version of the contract privately to arbitrators and jury members (by encrypting it using their public keys). Arbitrators and jury members will decrypt the plain English version of the contract and verify that its hash correspond on the one in the smart contract.

In order to avoid the ability to an outside observer to determine which smart contract have the same plain English version, each plain English version of the smart contract must include a random salt.

3.1.2 Requests

The arbitrated smart contract must include functions for parties to make a request to the court. A request would be in the form of an ABI bytecode (code which will determine which function and value of its arguments) that the court will include to a call to the arbitrated contract if the request is approved. The arbitrated contract must include functions for an opposing party to make a counter-request when a request is submitted to the court. Submitting requests or counter-requests will involve paying an arbitration fee to the court which will only be refunded to the winning party.

If no counter-request is submitted to the court (note that counter-request can simply be doing nothing), after a defined amount of time, the request is automatically granted. If a counter-request is submitted a dispute is created.

3.1.3 Random number created by two opposing parties

First instance arbitrators and juries (when appealed to a jury) are randomly selected. The Ethereum Virtual Machine is deterministic and does not include a resilient random number generator. Using the hash of a block would allow minors to censor blocks with hashes leading to arbiter or juries they don't want to be selected. In order to tackle the issue, a random number will be generated by the two opposing parties.

Again a commitment scheme will be used. The party making a request to the court (the first party) will create locally a random value and submit its hash with the request (the hash is a commitment to the random number). The party making the counter-request (the second party) will submit a random value with its counter-request. After this submission, the first party will reveal its random value which will only be accepted by the court contract if it matches the hash which was submitted. The random value will be computed by applying a bitwise XOR to the random values of both parties.

This scheme produces a random value as long as at least one party provides a random value. The parties could collude to produce any value, thus choosing the arbitrator and the (potential) jury members. However, since those parties are opponents, collusion is not a problematic issue.

Would the first party fail to reveal its committed value after a defined amount of time, the counter-request would be granted by the court.

3.2 Reputation tokens

In order to protect the system from a sybil attack(6), reputation tokens have a fixed supply. They would initially be given to people who have taken part in the crowdfunding of the decentralized court. A lesser part of them will be given to project contributors.

Reputation tokens will determine the probabilities to be drawn as an arbitrator or a jury. They would be valuable because arbitrators and jury members will be granted arbitration fees. Arbitrators and jury members behaving dishonestly will loose part of their tokens while those behaving honestly will win reputation tokens over time.

3.3 Court session

3.4 First Instance Arbitration

Using a jury system in first instance would cause scalability issues and an effect of responsibility dilution among the jury members (most jury members would assume that another member has to investigate the dispute and would not ask extra information to parties themselves). In order to avoid this issue, disputes are handled by arbitrators in first instance.

In order to be chosen as an arbitrator, reputation tokens holders will have to activate their tokens for each session of the court. This ensure that arbitrators are only drawn among active members of the court. The probability to be chose as an arbitrator will be proportional to the amount of tokens activated for arbitration. Since activating a high number of tokens will result in being given an high amount of disputes to arbitrate, reputation tokens holder may choose to only activate part of their tokens to avoid being given more cases than what they can handle. In order to prevent members from activating more tokens than they have or withdrawing token they could loose, activated tokens cannot be transfered.

After the random number of a dispute is created, an arbitrator is randomly drown. Parties will give him information about the dispute (this includes the plain English contract, see section 3.1.1). The opposing parties will be given a determined amount of time to produce pieces of evidence supporting their case. The arbitrator can ask for additional information. All those exchanges are encrypted using public key cryptography.

After considering all the elements of the dispute (and policies of the court, see section 4.2.1), the arbitrator give a ruling in favor of a party. If the loosing party does not oppose the ruling of the arbitrator, the decentralized court contract make a call to the arbitrated contract with the ABI bytecode specified in in the request or counter-request. The arbitration fee of the party winning the case is refunded and the arbitration fee of the loosing party goes to the arbitrator.

If the loosing party believe that the case was ruled improperly, this party can appeal to a jury. Appeals cost extra arbitration fees, therefore parties should appeal only if they believe that the arbitrator behaved dishonestly or made a mistake.

3.5 Jury System

Jury members also have to activate their tokens. However since being a jury member is a lesser commitment than being an arbitrator, the tokens only have to be reactivated after sessions where the members has been drawn. This way inactive jury members may be drawn once but won't be drawn again unless they reactivate their tokens.

The probability to be drawn as a jury member is proportional to the amount of activated tokens. Using the random number created by the opposing parties a segment of the jury activated tokens is drawn. The length of the drawn segment for the first appeal is specified by the court. Further appeals double that length (while doubling the cost of appeal). When the length of drawn token is superior to the amount of activated tokens (which means that all potential juries had to rule the case), appeal is not possible anymore. Token owner can be partially drawn. To show an example, we will assume that the random number for the dispute is $r = 59657401588$, the segment length is $l = 2000$ and we have 6 parties with a total of $t = 10000$ tokens following the repartition shown in figure 3.5.

The randomly drawn segment will be $[r \mod t, (r \mod t) + l[= [1588, 3588[$.

Token owner	Activated	Start	End	Drawn
A	1000	1	999	0
B	1500	1000	2499	912
C	500	2500	2999	500
D	3000	3000	5999	588
E	1500	6000	7499	0
F	2500	7500	9999	0

Figure 1: Example of activated token repartition

So owner C will be completely drawn, while owner B and D will only be partially drawn. The number of drawn tokens of an user will be their voting weight.

As with first instance arbitrators, parties will communicate information with jury members. After a defined amount of time, jury member will vote in two step. In the first step they will commit to a vote and in second step they will reveal their vote.

In the first step jury members will post a commitment of their vote by posting a hash of their vote concatenated with a secret value. It is important that votes cannot be known before they are revealed, otherwise jury members would be incentivise to vote the same way as previous votes. In order to avoid jury members to reveal their vote during the first step, anyone knowing the secret value of user will be able to submit it to the court contract and steal the tokens of this user while invalidating its vote.

In the second step, users will post their secret and their vote to the court. The court contract will verify that the hash correspond to the vote and secret posted by jury members.

In order to incentivize honest behaviors, jury members who had voted differently than the outcome or failed to reveal their votes will loose some part (for example $w = 0.1$) of their tokens which will be redistributed to jury members who voted like the majority.

In our example, let's assume that members B and C voted to grant the request while member D voted to grant the counter-request. The request will be granted (unless there is an appeal). Member D will loose 60 tokens, members B (respectively C) will gain 39 (respectively 21) tokens.

If the jury makes the same ruling as the arbitrator, the appeal arbitration fee will be splitted among jury members in proportion of their drawn tokens. If the jury makes a different ruling, the additional arbitration fee will be refunded to the appellant and the arbitrator will loose reputation tokens (having a value equal to the appeal arbitration fee) which will be splitted among jury member in proportion of their drawn tokens. It is important that the value of reputation tokens which can be lost by the arbitrator matches the value of appeal fees in order not to avoid jury members having game theoretic incentive toward confirming or overturning the verdict of the first instance arbitrator.

In future versions of the system, we plan to make juries vote whole ballots (including multiple disputes) instead of separated disputes. A machine learning method will be used to weight votes of the juries by a coherence of the vote of a jury with other juries. The method chosen may be an adaptation of the SVD-resolution algorithm described in the Truthcoin white paper (9). Or a method inspired by the work in the field of sensor fusion, as we could consider each jury member to equivalent (on an algorithmic point of view) to a sensor.

4 Improving the court

Those issues will be out of scope for the hackathon. However they would be dealt by the team in the future.

4.1 Improvement process

4.2 Sub-court system

4.2.1 Voting policies

4.3 Choice of arbiters

4.4 Disputes involving more than two parties

4.5 Lawyer system

4.6 Investigator system

5 Disclaimer

This is an alpha version of the white paper of the decentralized court project. It was created during the hack.ether hackathon within a limited period of time. Therefore, the design of the court is subject to changes and this paper may contain imprecise or even incorrect statements. If you have any comments or want to be part of the project, you can contact us on the Gitter chat at `https://hack.ether.camp/public/decentralized-court`.

References

[1] Bitrared. `https://www.bitrated.com/faq/`.

[2] Gnosis. `https://gnosis.pm/`.

[3] Oraclize. http://www.oraclize.it/.

[4] Buterin, V. Decentralized court, post on ethereum reddit. https://www.reddit.com/r/ethereum/comments/4gigyd/decentralized_court/.

[5] Buterin, V. Schellingcoin: A minimal-trust universal data feed. https://blog.ethereum.org/2014/03/28/schellingcoin-a-minimal-trust-universal-data-feed/, 2014.

[6] Douceur, J. R. The sybil attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems* (London, UK, UK, 2002), IPTPS '01, Springer-Verlag, pp. 251–260.

[7] Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf/, 2008.

[8] Peterson, J., and Krug, J. Augur: a decentralized, open-source platform for prediction markets. http://bravenewcoin.com/assets/Whitepapers/Augur-A-Decentralized-Open-Source-Platform-for-Prediction-Markets.pdf/, 2015.

[9] Sztorc, P. Truthcoin, peer-to-peer oracle system and prediction marketplace. http://www.truthcoin.info/papers/truthcoin-whitepaper.pdf/, 2015.