Decentralised Governance and Dispute Resolution in the Digital Era

Lucas Oliveira

November 2019

Abstract

As social media grows, the content inserted on it surpasses millions of billions of data, and unfortunately, the review of their contents is debatable, even by the companies holding the rules of content review. We can see many users complaining that their content was demonetized, deleted, or their account blocked on various social media platforms. At the same time, news websites, blogs, and other websites that have a commentary section have a hard time reviewing and blocking texts that contain hate speech, misleading context, and other problematic content. The system is flawed, and the ecosystem of the growing internet can't rely on old ways to solve disputes. When we expand this vision, we can see that not only social media but any system that relies on dispute decisions based on certain rules, such as a judgment system, faces problems with the decisions made, the time consumed to reach a decision, and the acceptance or rejection of the rules by the involved parties.

When I started to write this white paper, my main goal was to create a distributed approach that could help companies expand their review process without the need to hire thousands of additional workers to review an exponential amount of data, which will be almost impossible for them to handle in the future.

However, as I explored this idea further, I realized that the consensus method I was trying to achieve could create a new and private judgment system.

Therefore, in order to elevate content review, games, and even the justice system to a new level, a new approach to this issue needs to be pursued.

1 Introduction

The goal of settlementIO it's to create a DAO(Decentralized Autonomous Organization) using a Level 2 settlement algorithm described on this paper , being able to create a Concurrency based on smart contracts where the dispute will be issued, reviewed and the result recorded , using digital signature, third party reviewers and law abide owners. To achieve this Goal , the Level 2 settlement algorithm that contentIO will use it's called Punitive Proof-of-Adequacy

2 Punitive Proof-of-Adequacy

The punitive Proof-of-Adequacy is a type of "Layer 2" settlement algorithm protocol that operates on top of a blockchain-based smart contract network that aims to use distributed consensus to achieve a result on a **settlement** dispute. In the PoA protocol, there is an earlier level of consensus where the peers who hold the PoA act, creating a network channel (a Pool), assigning an odd n P2P curators (*Reviewers*) connected to the network, giving them access to a **settlement** dispute via various combinations of random selection of *Reputations*, thus generating a smart contract of the result of the dispute, using the fees to be shared among peer curators and the creator, known as *Witness*, of the next block.

In order to record on a blockchain, PoA has little to no participation, and the *Witness* can be any Index smart contract-based network.

2.1 Process

When a *Reporter* files a **settlement** against the *Appealer*, the Issuer (I) receives the dispute, then sends the **settlement** to be reviewed (r). At this moment, a *bounty* can be issued. It is at the discretion of I to notify the *Appealed* of the **settlement** dispute in progress. If not, the information must be sent. For example:

- settlementIssued

```
 {
    data : hash_1,
    issuer: hash_2,
    reporter: hash_3,
    appealer : hash_4
}
```

The request is sent to an Assignment Pool where the Issuer will accept a number (n) of r. n is an odd quantity, with a minimum of 7 and a maximum of 25 of r. The r assigned to the content to be reviewed are designated based on the following:

- Position on the assignment pool
- Reputation
- Waiting time

After the designation, the content receives feedback from the n assigned peer R, and the final *Result* is settled. The *Result* is then sent back to the PoA algorithm to record it on a blockchain.

The PoA sends the data to a pool of results to be recorded on a blockchain. The data must contain:

- The reference/data of the settlement Result
- The price distribution among the winners of the voting settlement, adding 60% of the issued bounty and subtracting the Witness fee

  $$G = pd + \{\, b \cdot [b \neq 0] \cdot \frac{60}{100} - wf$$

  , if any.
- The Witness Total settlement fee attached to 40% of the issued bounty

  $$SF = wf + b \cdot [b \neq 0] \cdot \frac{40}{100}$$

  , if any.
- The hash of all reviewers who participated in the settlement
- The hash of the writing Witness

```
settlementIssued = {
    data : ‘<SETTLEMENT PROCESS>‘,
    issuer: ’688A044D54361D5762100BD1E6559AF4’,
    reporter: ’4A12FE7C3773A2B801BFFEB341A77949’,
    appealer : ’39AAF3FEF253178E15963B9CC27DA138’
}
```

G is the total grant of the **Result**, pd represents the price distribution, b stands for the bounty, SF represents the settlement fee of the **Witness**, and wf represents the fee of the **Witness**.

A **Witness** is assigned to generate a block on the blockchain via PoS, PoW, or the **Witness** Assignment Logic. Then, it will take n data from the **pool of results** and record it on the blockchain.

2.1.1 In a nutshell

1. A dispute is made to the network following a certain rule:

   $$Dispute \rightarrow Network$$

2. One peer of the network, defined as the Issuer, will register the dispute and request to other peers to be assigned:

   $$Issuer \rightarrow OtherPeers(AssignmentRequest)$$

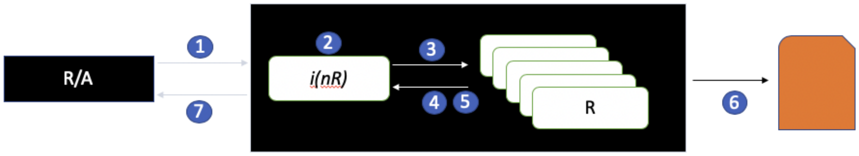

Figure 1: Parent block

3. Peers request to assign as a Reviewer to the dispute, and an odd number of Reviewers will be chosen by the issuer:

$$Peers \rightarrow Reviewers(AssignmentRequest)$$

$$Issuer \rightarrow Reviewers(SelectionofOddNumber)$$

4. Peers granted availability will be sorted by their reputation on the network:

$$Peers(Available) \rightarrow SortedPeers(SortingbyReputation)$$

5. Reputations are defined based on previous actions the peers made:

$$Peers \rightarrow Reputation(BasedonPreviousActions)$$

6. The result generates a Smart Contract:

$$Result \rightarrow SmartContract$$

7. The dispute results are informed back:

$$Result \rightarrow InformIssuer$$

Note: The lexical representation captures the relationships and actions involved in each step of the process, but it does not provide specific mathematical equations or formulas.

Here's the provided section formatted into lexical representation:

2.1.2 Appeal

An appeal of the Result can be issued, restarting the process and generating new data to be recorded on the block referring to the hash of the last Result.

If the result is the same, the settlement is closed and it cannot be appealed again (appealFowardResult):

$$result_1 = result_2settlementclosed$$

If the appeal yields a different result, but the algorithm requests another appeal process to retrieve the result of the Issuer, the punitive algorithm is triggered. This algorithm reduces the Reputation and charges a fee to be deducted from the losing peer reviewers, adding the charge to the prize of the next settlement dispute of the Issuer (appealBlockingResult):

$$result_1 \neq result_2 punitivealgorithmtriggered$$

The punitive algorithm involves reducing the Reputation (reducedReputation) and charging a fee (fee) to be deducted from the losing peer reviewers:

$$reducedReputation = currentReputation - reputationReduction$$

$$nextSettlementPrize = currentSettlementPrize + fee$$

where currentReputation, reputationReduction, currentSettlementPrize, and fee are variables representing the respective values in the algorithm.

The charged reviewers cannot appeal as they do not have the required level of permission among the three peer-to-peer (p2p) actors.

appealFowardResult

```
{
  LR: hash1,
  RR: hash2,
  G: pd + (b \cdot (60/100)) - wf,
  SF: wf + (b \cdot (40/100)),
  RW: [hash3 , ... , hashN],
  W: hashN
}
```

appealBlockingResult

```
{
  WR: hash1,
  LR: hash2,
  RR: hash3,
  G: pd + (b \cdot (60/100)) - wf,
  SF: wf + (b \cdot (40/100)),
  RW: [hash4, ..., hashN],
  W: hashN
}
```

Here, the values for $hash_1$, $hash_2$, $hash_3$, and so on, are the SHA256 hash numbers associated with each key in the ‘appealBlockingResult‘ object.

2.1.3 Assignment pool

When a R requests a **settlement** for a review, the Issuer will get the first reviewer available on the Pool (P) of assignment and ask for a review, then the reviewer is sent to the end of P. The position on the assignment pool is defined by the time connected to the network, and the positions are distributed among the peers. Having the top position on the Assignment Pool doesn't automatically grant the reviewer the assignment from the Issuer; it's only another measurement used to define the assignee of a dispute.

$$_A =_I .get((reviewer) \Rightarrow reviewer.position$$

2.1.4 Assignment Ordination

In order to keep the assignment pool with the most trustworthy results as possible, the reputation of n reviewers, denoted as R, should be differentiated. This allows for the advantage of distributing n assignees on a single review by a range of experienced reviewers to new ones (or those with poor trustworthiness) in the peer network. The range is defined by the reputation of the reviewers, which is explained in the section on Reputation.

To control and maintain the trustworthiness of the network, the votes will be weighted by the assignee's reputation. This weighted voting system ensures controlled and improved results for the reviews. The objective is to avoid misleading engagement or malicious behavior from more experienced peer users, prevent power centralization among highly reputable peers, and provide a more honest result for the review.

The weight of the votes is discussed in the section on Vote System Weight.

Please note that references to other sections such as "Reputation" and "Vote System Weight" should be adjusted based on their actual section numbers or titles.

2.2 Bounty

Bounty it's a additional fee inserted on the **settlement** dispute , created to enable the possibility of a Issuer to require a faster result on a dispute it's generated. A dispute with an added bounty it gains priority on top of any other dispute on the network.

2.3 Reputation

A range of reputation (rp) is assigned to the reviewers (R), which includes the categories "Trustful," "MidLevel," "Non-Trustful," and "Undesirable." When a new reviewer is created, they are initially assigned a "Non-Trustful" level of reputation. The reputation level can change based on their engagement and performance within the system.

2.4 Waiting time

While in the pool, the reviewers (R) will record their waiting time, which represents the duration they have been connected to the network without being granted an assignment. The waiting time serves as a metric when an Issuer receives a request for assignment from the reviewer peers (R).

Certainly! Here's the formatted section in LaTeX:

2.5 Vote System & Weight

The reviewers (R) will receive the settlement and the rules defined by I and determine whether the settlement is in favor or against the reporter. After the reviewer gives their vote, I will request the next assignee reviewers (R) from the pool (P), with each reviewer having a different level of reputation as defined in the Assignment Ordination section. Once a minimum of 7 votes is reached, the algorithm checks if the settlement has a result. If not, it will assign another reviewer until the next odd number of votes is reached, and this process continues until a maximum of 25 votes.

The pool (P) is responsible for queuing the assignees for the next settlement. The quantity of reviewers (R) to be assigned (A) will be a minimum of 7 and a maximum of 25.

The assignees (A) must be distributed among the following reputations:

$$r_{trust} = 0.25 \times |A|, r_{mid} = 0.30 \times |A|, r_{nont} = 0.45 \times |A|$$

The reputation determines the voting power of each assignee. The "Non-Trustful" (ntt) serves as the benchmark voter, with the "MidLevel" having 1.5 times the voting power of ntt and the "Trustful" (r_{trust}) reviewers having 2.5 times the voting power of ntt.

2.6 Result

The result of a dispute will generate a Smart Contract to be recorded.

2.7 Pool of Results

All the results will be queue on a pool to be consumed.

2.8 Reputation gratifications/punishment

After the result it's defined , the Reputation algorithm it's fired, analysing the voters and the outcomes of the result. When analysed, the algorithm it grants the gratification, which is gave by taking the total fee and spreading the value to the winner voters and the amount divided by the peers Reputation.

2.9 Benefit of doubt

A **settlement** dispute must have a trustful result, and for that, a **settlement** has to be reviewed by an odd n of **Reviewers** (R). To ensure a result is trustful, the algorithm allows the **benefit of doubt** of any **settlement** result, and this doubt is granted to the 3 peer-to-peer (p2p) actors. They are the two parties in the dispute, the **Reporter** and the **Appealer**, and the law-abiding **Issuer**, who defines the **rules** where a **settlement** dispute is taking place. The amount of **benefit of doubt** is determined by its own cryptocurrency blockchain network, but in order to avoid fraud, it is recommended to follow the instructions in this paperwork.

3 Rules of the settlement

The rules of the settlement are defined by the **Issuer**, so the Proof of Authority (PoA) designates this actor as the single source of truth when a **Benefit of Doubt** is present.

Sure! Here's the formatted text in LaTeX:

4 Actors

4.1 Issuer

It is the peer who has access to the Proof of Authority (PoA) network to send the **Settlement** dispute over the pool of assignment and accept Reviewers' requests. The Issuer can be either the Reporter, Appealer, or a network access holder, such as companies who wish to create their private PoA network.

4.2 Reporter

The one who generates the dispute to be distributed by the Issuer.

4.3 Reviewer

The first n users connected to the platform who self-assign the review lock the analysis of that content and give their result. The distribution of the review is handled by the Witness, who assigns the review to the pool and checks the Reviewer's proof-of-integrity based on the Issuer's internal policies. The users cannot know each other, so the system works with non-trustful nodes.

4.4 Witness

The Witness is the algorithm (miner) that holds the distribution of the Issuer's content for review through the Reviewers, defined by location, reputation, and proof-of-stake. Once the reviews are issued, Witness algorithms take the reviewers' results and record the content result on the Chain, working as a notary book for consultation by the Issuer. Therefore, the prize value is divided between the Reviewers and credited to their accounts, which will be paid monthly/weekly. A Witness node that inserts the Review first is granted the fee charged from the Issuer. The Issuer can also be a Witness for the chain.

4.5 Appealer

Users/reporters who have a settlement result against their will can appeal the result of the review, and a new request is issued. n appeals can be granted by settlement, defined by default or by the Issuer. It is at the discretion of the Issuer to notify the Appealer of the ongoing settlement dispute, but the Appealer must be notified after the result. The Issuer must also inform the network if the Appealer was notified when the settlement dispute was posted.

5 Security Measures

How to secure the block.

6 Pros

Companies can use the platform where the pool of nodes to be reviewed can provide more returns for the content Reviewers. The platform can be deployed on-premises or on the cloud. Reviews can be more reliable.

7 Cons

Please note that the formatting and structure of the text in LaTeX may vary depending on the specific document class and packages used.

8 Conclusion

In this paper, we have presented a decentralized dispute settlement system based on the Proof of Authority (PoA) network. The system allows for fair and trustful resolution of disputes by leveraging the power of peer reviewers and the consensus mechanism.

Through the use of assignment ordination and reputation-based voting, the system ensures that the settlement results are reliable and unbiased. The involvement of the Issuer, Reporter, Reviewers, and Appealer in the dispute resolution process provides a robust framework for addressing disputes effectively.

The security measures implemented in the blockchain network provide a high level of trust and integrity to the settlement process. The use of cryptographic hash functions ensures the immutability and authenticity of the data stored on the chain.

Overall, the proposed system offers numerous benefits, such as increased reliability of reviews, transparent dispute resolution, and the ability to create private PoA networks for specific industries or organizations. However, there are still challenges to overcome, such as scalability and ensuring the participation of a diverse pool of reviewers.

Future research can focus on addressing these challenges and further enhancing the security and efficiency of the system. By continuously improving and refining the decentralized dispute settlement system, we can create a more trustworthy and reliable environment for resolving disputes in various domains.

Acknowledgments

We would like to express our gratitude to the individuals and organizations who provided support and assistance in the development of this research. Their contributions have been invaluable in shaping the ideas presented in this paper.

References