N • E • R • V • E

A Decentralized Task Validation System

Abstract
A purely decentralized method of validating the proper execution of an arbitrary task would allow the online community to assign tasks to each other and associate payments without the need of using authoritative third parties. The idea of a decentralized democracy provides part of the solution, but its complexity exceeds the current scope of the blockchain ecosystem. We propose a solution to the complexity problem using limited binary cases. Tasks are voted on by participants and are either deemed valid or invalid. To increase reliability of task validation we target the NERVE application area to digital environments with high reliance on reputation.

1. Introduction

Live streaming services are growing fast and are highly influential on the gaming and adult entertainment industry. With the advent of augmented reality devices we are also entering an era of public IRL (abb. „in real life“) streaming. Most streamers rely on donations or third-party payment systems with high fees. There are no direct and private payment channels between viewers and streamers. Donations are always given without any condition and most subscription systems are only linked with minor benefits. The adult entertainment industry is offering exclusive features or direct communication with the streamer for a fee, but most platforms charge a 50% or even higher fee to the streamer. On the most popular live streaming platforms (YouTube, Twitch) it is not possible for the viewer to set any incentive that would change the course of the stream. We propose NERVE to create a new method of fundraising for streamers and to pose the opportunity for viewers to change the course of streamed content in real time. Our system allows the user to assign a task to a streamer and associate the fulfillment of the given task with a direct payment to the streamer.

2. Voting

The blockchain technology is most often used to establish direct and private payment channels between two parties. We use the Ethereum blockchain and the Ether currency to facilitate this function in our NERVE ecosystem. A task can be initiated by and assigned to any Ethereum address and has to contain some Ether as payment. The staked Ether are distributed to the target address if the task is fulfilled or are refunded to the initiating address if the task is deemed invalid. To discern whether a task was fulfilled or not we establish a decentralized and democratic decision-making process in our smart contracts. Each participant in a task has to stake at least the same amount of Ether to acquire voting rights on a certain task. If the majority of participants vote in a positive manor, the task is deemed fulfilled and all funds are distributed to the task executor. This voting process can only consider single task propositions and the outcome may only be binary, i.e. the task is either fulfilled or not without the possibility for third options.
NERVE enters into the territory of decentralized democracy, which is a highly innovative form of using blockchain technology. Therefore, we are employing a cautious approach with regards to the implementation of more complex functionality. NERVE will be one of the first working products in this new area of blockchain functionality. As a starting point, it is in our intention to approach this topic as simple and direct as possible. Our system could be used in many more environments, but we think that the live streaming industry delivers the most reliable data to refer to as a main target sector.

3. NERVE Core Contract

The NERVE Core contract facilitates the possibility to assign an arbitrary task to any party present on the Ethereum blockchain. To ease the process of task assignment the contract integrates a name registry allowing users to link a unique name to their address. This is particularly useful for social media environments. The relation between public address and registered name is public and can be reconstructed at all times without evoking GAS costs.

As by the initiation of a task, any user represented by an ETH address can either take on one of the two following roles:

- - The challenged user, which is a unique role being required to perform a task by participants

- - A Participant offering a stake to the challenged user to perform the assigned task
  To initiate a task, a participant needs to know to whom they would like to assign the task. This can either be done by entering a public address or by using the corresponding name from the name registry. In addition, the participant broadcasting the task has to send a description, duration and a minimum stake along with the task. Each task and its contents are saved on the blockchain and labeled using unique ID.

Description

The description is a String consisting of 8-64 characters that describe the task. As this contract acts without any authority, the content of the description can be chosen freely and will never be censored or limited in any way.

Duration

The duration is an Integer defining the time remaining until the task closes in seconds. Every task should be fulfilled during this period, otherwise it may be deemed invalid depending on a vote casted among participants. Subsequent to the duration, an additional time amounting to 50% of the duration is added to allow voting on whether the task has been completed satisfactorily.

Minimum stake

To be able to broadcast a task to the blockchain, a participant will be required to send a freely selectable amount of ETH along with it. The ETH are linked to the user's public address and remain locked through the contract until the voting period has ended. Once the task is fulfilled, a popular vote among participants is casted to evaluate, whether the task has been completed satisfactorily. Depending on the result of the vote, the ETH are either reallocated to the challenged user or unlocked for participants. Should the ETH be unlocked, participants are able to withdraw the full amount of their stakes.
If an additional participant wants to join an existing task, they are required to send at least the corresponding minimum stake of that task. This ensures that each participant assigns at least the minimum stake to a task. Each participant who has stakes in a particular task is allowed to participate in the voting process and is given one single vote.

4. Possible Attack Scenarios

In the following we outline three ways this system could be exploited and how we prevent these from happening. The attacks are defined as follows: False task fulfillment, false task rejection and reputation farming.

False task fulfillment

The challenged user has put stakes in a task that was assigned to himself. He or she then uses his voting power to deem the task as fulfilled without actually completing it. This scenario is very unlikely to happen in an environment, where the player is live-streaming and the outcome of his or her performance is easily verifiable. It would be obvious that the player did not fulfill their task and he or she would lose their credibility in front of their users. We advise participants to choose stakes depending on the reputation of the challenged user.

False task rejection

The challenged user has fulfilled the task but the users (or a single user with many addresses) vote against him or her. As the player would not receive their compensation, we deem this attack as the most critical. To cope with that circumstance, we instantiated a reputation system for participants. Every time a participant approves a completed task through their vote, it will be checked, whether the participant voted with the majority or minority.

If within the majority, the participant receives a positive reputation gain of one reputation point. Every time a participant votes against the majority, they will lose half of their accumulated reputation (independent of the outcome of the task). As the voting system is binary (it only considers simple yes or no cases), fraudulent activity should be obvious and easily observable. To ensure success, a potential attacker would need a 51% majority against honest voters, while being at risk of losing all of their stakes. Challenged users are therefore recommended to only take action on a task provided the average user reputation is high or many of users are participating in a particular task.

Reputation farming

Introducing the above-mentioned reputation system the risk of deliberate reputation farming and subsequent sale of high reputation accounts emerges. As this would likely happen in a closed eco system with the fraudulent user making up both the participants as well as the challenged user, there would be no risk of losing stakes except a transaction fee as charged by the Ethereum network. This transaction fee may be negligible or significant, depending on the amount at stake. To further prevent reputation farming from happening, a 10% fee on distributions to challenged users is introduced. This will ensure that reputation farmers would incur significant losses. As an additional feature, and by rule of the Ethereum network, all participants and their votes on specific tasks are public and it is easy to identify fraudulent parties. Besides being at risk of losing their reputation very quickly by voting against the majority, we trust that the social media environment would identify and blacklist fraudulent users on their own.

5. NERVE Bets Contract

The NERVE Bets contract is subordinate to the NERVE Core contract and is supposed to enhance its user experience and user spectrum. Nevertheless, this module is capable of running on its own and has many niches to fill. It allows an authoritative party (e.g. the player) to instantiate their own betting office. Everyone is able to post a bet to the blockchain and users can then decide to bet for or against it. Each bet is labeled with a unique ID and the address and name of the initiator always remains public, while the users and their bets stay internal and are not accessible. The contract automatically determines the bet quotas depending on the relation of the incoming bets. As soon as the authoritative party decides to close the bet and gives an official result, all collected funds are distributed evenly between the winning users.

This system is mainly meant to enhance the user experience of NERVE Core, but we allow the authoritative party to publicly determine their share of each bet on their own.

As this module only facilitates betting on simple binary outcomes (yes or no), it should be very easy to identify fraudulent behavior of authoritative parties. In the end it is important to state, that this system is based on a centralized decision-making process and each user has to decide by themselves whether they trust the authoritative party.

6. NERVE Equity Token

All revenue shares of current and future NERVE modules will be transferred to the NERVE Equity Token contract and are evenly distributed to all token holders and their respective share of NERVE tokens. Each token holder is responsible for redeeming their shares – there are no shared GAS costs. The contract allows for dynamic “dividend” redemption, meaning there is no time limit or minimum “dividend” requirement involved. As all NERVE contracts, the NERVE Equity Token contract is publicly available and can be easily verified.

All NERVE contracts are static and do not involve ownership of any person over the contract!

7. Decentralized Application (DApp)

The open source structure of the NERVE contracts allows any third party to create their own platform solution and decentralized application without asking for permission. This allows the NERVE ecosystem to expand and adapt to future markets without friction.

Parallel to the open structure, we provide our own DApp that will allow users to assign tasks to players and place bets on top of them. We are one of the first in the blockchain space to utilize Unreal Engine 4 for our products and will support Windows, MAC, Linux, Android and iOS operating systems.