Decentralized Survey Application Proposal

Team Bubble: Raafay Alam, Kamil Balitzki, Carolin Stein, Piotr Witkowski

Motivation & general idea

Our idea for this project is to create a decentralized survey application using a blockchain, to ensure the integrity and completeness of the survey's data.

Nowadays there exist multiple online survey applications that facilitate the process of bringing together researchers and participants, offering the possibility to upload surveys which can be completed by a pool of agents, who then will be recompensed for their work. The integrity of this data is vital for the work of researchers or whoever is relying on the results of the survey. It is therefore of great value to be able to proof, that the data has not been edited by someone else but the participant in any kind of way: no record should be removed or added after the initial poll and given answers shouldn't be modified. In today's research, the assumption of the correctness of the data is mostly based on trust in the researchers. Especially when thinking of outsourcing surveys to other institutes, trust might not be sufficient anymore: We believe that using a blockchain-based approach could solve this issue. Agents can register to our survey application, receiving participation request for a survey with a certain compensation. When agreeing to the offer, a smart contract will ensure the conditions of participation are met by all parties: The participants will submit their answers and receive the compensation as specified upfront. The contract will store the array of answers on the blockchain in a secure encrypted manner, so the volume of records can be proven. The client will receive the keys to decrypt the survey results and can, if needed, proof at any time that he used the original data.

Technical architecture (overview)

The application will be based on Ethereum using Smart Contracts. Access to the application could be gained via a Website or other GUI which could be implemented by a third party. It will be possible to submit or participate in a survey and purchase results. For usage of those features, a small fee will be paid to the authors of the application.

To submit a survey GAS for processing will be paid. Besides, the duration of the survey and the payout model will be part of the survey description, e.g. the shares of the payout upon result sale or pre-paid participation fees and no result sale. Participation in a survey will also require the payment of GAS. This will be done to raise the bar for misuse and enhance result quality. Answers to survey questions will be encrypted and stored on the blockchain. After a survey has ended, the decryption key can be retrieved as specified by the payout model chosen by the survey initiator.

Technical challenges & solutions

- Payout models need to be specified in a smart contract conformant manner
- Key storage for public and private key needs to be solved
- Survey answer validation/assessment could be implemented in the smart contract:
    - not sure yet about this, may be beyond the scope of our application
- Agents should only see what they're supposed to see:
    - survey description public in the blockchain

    - survey data should be encrypted on the blockchain
    - organizer/ buyer of the data gets the key to decrypt data
    - participant addresses should be anonymous*
- How to prevent that people participate multiple times?
    - Participants pay fee upfront so participating many times at once might be costly
    - Ethereum accounts could be hashed and stored to the blockchain with the survey name to *ensure anonymity and one-time participation

Our implementation schedule proposal

20th of May: get application proposal accepted

Implementation:

- 1st week: First quick-and-dirty prototype
    - survey submission
    - participation
    - result retrieval
- 2nd week: additional main features
    - payout model
    - encryption/ decryption
    - “basic” website
- 3rd week: nice to have features
    - validation submission
    - improved website
- 4th week: around (20th of June): final touch
    - bug fixing
    - writing tests?
    - project submission

8th of July: project demo