Human ATM

Cloza, Nicolás; Mendiola, Julián; Rosas, Agustin; Tallar, Nicolás

July 23, 2017

1 Introduction

With the great widespread of usage ATMs, shortage of cash on them and of ATMs themselves (such as in remote locations) has become a common problem. Both of them, frequently cause bank customers problems, due to their lack of time or lack of nearby ATMs. Also increasing in usage are blockchain-based technologies which is driven in part by the transparency and safeguard properties provided by them, allowing an easy and public audit of the operations being carried out. This technologies not only enable carrying out simple money transfer transactions but also decentralizing and increasing the transparency of code execution, as is the case with the *Ethereum*[1] cryptocurrency.

HumanATM is a project that, driven by the mentioned advantages of the blockchain technologies, solves the mentioned customer problems that arose with the increase in use of ATMs. Similarly to how *Uber* solves the commuting problem by allowing anyone to offer its service, *HumanATM* attempts to solve the shortage of cash (caused by either shortage of cash on ATMs or ATMs themselves) by allowing cash transfer between any two bank clients. So, if a user is in need of cash, while another one has extra cash, the latter one can use *HumanATM* to give it to the former, in exchange for tokens. This tokens will end up being exchanged for the same amount of money that was given (with an extra amount as a reward), but this can be in the future be extended for allowing transfer of tokens between users and even exchanging them for prizes.

For allowing, this *Ethereum smart contracts* were used, so that all operations regarding token distribution, transfer and exchanging tokens for money are registered in the blockchain.

2 Implementation

There are three user-related operations used in *HumanATM*, which are: user registration, token transfer and accounts reconciliation. In each of them tokens will be moved between or assigned to them, which will be of different categories: credit tokens or prize tokens.

2.1 Credit and prize tokens

As different bank clients will fit into different profiles (due to yearly earnings, current savings, among other characteristics), the amount of cash that they should be allowed to *"extract"* using *HumanATM* will be different. In order to be able to register this, users will only be enabled to receive cash if they have proper authorization, which will be represented by having each user *credit tokens*. So, when a certain user receives cash, this tokens are transfered to the previous owner of the cash. At this moment, the tokens become *prize tokens*, which can be later exchanged for money corresponding to the amount of cash transfered plus a reward.

So, as described in the example, *credit tokens* can only be used for transferring to other users, while *prize tokens* can only used for redeeming the money transfered plus the reward.

Both this tokens will be *simpler ERC20*-compliant[2], so they will follow the specified interface.

2.2 User registration and credit increase

As described in the previous section, users are assigned *credit tokens* (depending on their profile) which can later be used by them to receive cash from other bank clients. Due to this, registration

[1] Ethereum website: https://www.ethereum.org/

[2] EIP-179: https://github.com/ethereum/EIPs/issues/179

of the users to the smart contract is needed, so that during this registration an initial set of *credit tokens* is assigned to them.

As operations are being carried out by the user, this initial credit is depleted, so a separate functionality was added so as to increase this credit. This service could be used either automatically, periodically or upon request by the user.

As both operations involve changing critical user information, they can only be carried out by an authorized agent (the *owner* of the contract).

2.3 Token transfer

Whenever two bank clients decide to exchange cash for tokens, this operation is done using the smart contract, not only for correct token management but also for registering that this transfer took place. This enables any of the parties that took part of the transfer to audit it and to use it as a receipt of the transfer (possibly to use it as part of a complaint in case of any problem).

As previously explained, the user giving the cash will receive prize tokens, while credit tokens will be subtracted from the user receiving the cash.

2.4 Accounts reconciliation

With the current functionalities (which could be easily extended in the future) prize tokens can only be used for exchanging for the money corresponding to the cash given plus the reward. This should be done for all users periodically (twice per month for example), and the *smart contract* should be able to return the number of tokens owned by each client (so as to be later used to determine the money to be given to the clients) and also take the prize tokens from them.

Both this two operations should be done atomically, as if not the amounts own by the clients could change between them, resulting in invalid information being used.

3 Future updates

- **Token transfer not being paid by the user:**

  With the current implementation, the bank clients need to have ether in their account in order to be able to send the transactions for transferring tokens to other users. However, in the common scenario that the user only uses his *Ethereum* account for this application, it would not have any ether available to pay for the transaction cost. For this reason, it is desirable for them to not be in charge of paying for this transactions.

  One way this could be achieved is by having the backend (on the server) in charge of sending this transactions, which would do so upon receiving a message signed from the sender. This message should at least have the receiver of the tokens and the amounts transfered. However, if not done carefully this could introduce security problems such as a potential replay attack.