CiviGo

A distributed oracle technology to BETHer our world

Davi Bicudo	-	davig@ethz.ch
Mario Stöckli	-	stmario@ethz.ch
Joël Lindegger	-	lijoel@student.ethz.ch
Davide Bernardi	-	berndavi@student.ethz.ch

All members contributed equally to this report

Table of Contents

Abstract

During the BETH 2019 Hackathon challenge we created a concept based on blockchain to incentivize people to improve the local community. The project name “CiviGo” was created, which is a concatenated word from Civil Service and “Go”.

Due to the limited time of two days we focused on creating the basic model and on the modularity of the concept such that CiviGo can be easily used to integrate it into the existing Finance 4.0 platform of the FuturICT group. CiviGo is a sensor-based token-obtainer which uses two tokens and IoT-devices for validation and user convenience.

Introduction

Problem Description

Today's most local community problems, such as trash lying on the ground or a broken bench, are solved by the state civil service or aren't solved at all. Fortunately, the Swiss state civil service is quite efficient on solving these problems but sometimes it gets delayed and it still costs money. In other countries the state civil service is merely present, or some issues won't get solved at all.
If such an issue occurs there will be three kinds of people: people who simply don't care at all, people who complain and do nothing about it and people who try to fix the problem. Sadly, the people who try to fix it are the minority and people who complain and do nothing are the majority. But why are these people the majority? We simply think that there is a lack of motivation, why would I fix an issue if no one would recognize it? And, complaining is easier than fixing the problem or proposing a solution.
Our motivation is to create a system such that there is no lack of motivation anymore. We want to incentivize the people and recognize their work and contributions to the local community. We hope that maybe one day such a system will not be needed anymore because this beneficial behaviour will be seen as cultural behaviour.

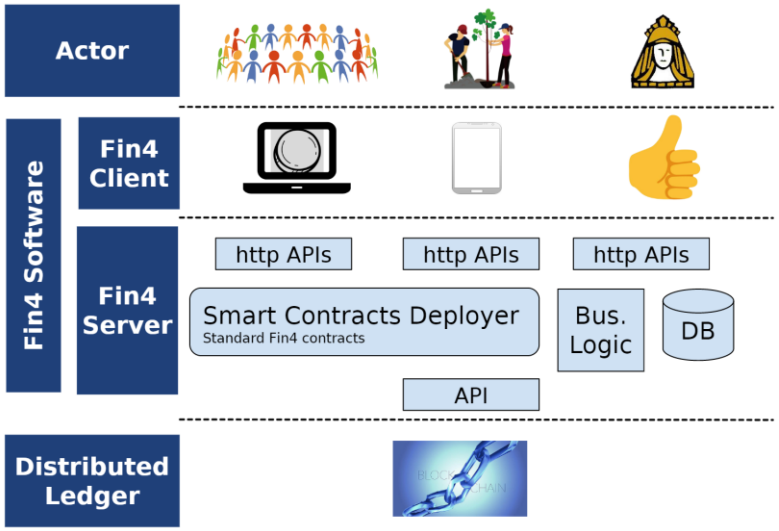
Finance 4.0

BETH 2019 challenged the attendants to construct ways how blockchain, smart contracts and the Internet of Things can be used to achieve the United Nations objectives of a sustainable world:

The attendants received up to eight challenges and we decided to participate on the first Finance 4.0 challenge. The goal of this challenge is to implement a token obtainer, which utilizes at least two tokens and one or more sensors. It also must be integrated in the currently available Finance 4.0 architecture.

The Finance 4.0 architecture enables communities to value a beneficial behaviour on a platform. The platform has three actors: the community, the oracle and the people who perform a task. People who perform a task can post their action on the platform in form of a text, picture or video. But before it gets published on the platform, the oracle must first approve it. Currently, the oracle is an admin of the platform. Once the action gets confirmed and approved, it gets published and the community can like this post. With every like the task performer will get a token, such that the task performer gets incentivized.

The Finance 4.0 architecture has three system components: The Actors, the Finance 4.0 software which consists of the Finance 4.0 client and server and the distributed ledger.[1]



As described above the actors are the community, the oracle and the people who perform a task.
The people can access the platform via mobile app, the Genesis App, or via the Web App, which are the Finance 4.0 clients. The Web App is written in ELM and the mobile app is currently available on Android. To confirm or approve a post, the admin must log in to the Web App since the oracle is only implemented in the Web App.
The HTTP API connects the Finance 4.0 clients with the Finance 4.0 server. In the Finance 4.0 server we have the smart contract deployer, some bus logic and the database. The smart contract deployer uses the Go-Ethereum client to deploy the tokens to the blockchain. The database which is written in MySQL is here to store the posts, manage the accounts and to manage this data.
And again, there is an API which connects the distributed ledger with the Finance 4.0 server. Currently, Ethereum is used to realise the smart contracts. For local purposes, Ganache is used and for global and for a testing network, Rinkeby is used.

The current implementation of the Finance 4.0 will help us a lot to realise our main concept, but we still must make changes in the client and, we need to modify the smart contracts. Also, Finance 4.0 has two main problems which are needed to be solved.

[1] *Finance 4.0 Platform & Architecture Presentation Slides - Mark C. Ballandies*

The Oracle/Proof Problem of Finance 4.0

Decentralization is one of the main reasons why the platform is based on blockchain. The core problem of Finance 4.0 right now is the oracle. The oracle decides which post will be published and the oracle is simply an admin, a human being. This human being decides with his own beliefs if the task performer has done a beneficial behaviour. With this action, the whole platform becomes centralized again, which is not our goal.
In our concept, we'll introduce a decentralized and democratic idea to solve this problem.

Spam Problem

The spam problem follows from the oracle problem. The oracle needs to approve or disregard every submitted post. What if many people submit many posts? Currently, the oracle must check every post. Additionally, one can submit many posts which are fraudulent. This is not scalable.
One solution can be that multiple oracles are allowed, but this solution would only postpone the scalability problem.
To avoid this, we propose a staking system and a limitation to the number of proposals.

Conceptual Model

Mission of the CiviGo platform

The CiviGo platform has its reason-to-be attached to the different communities using it. This may be a neighbourhood or a large city, but are typically associated to sharing physical space, although in principle the concept could be applied to virtual communities.
Through the platform, users can gather attention to important issues within the community such as a trash, graffiti, a broken bench, a polluted pond, etc. and collectively, in a decentralized manner, discuss the problem and implement solutions.

The CiviGo process

The chart below describes the process of interaction of users in a community within the CiviGo platform.

Even though there is a different term for designating users participating in each phase, they are not necessarily different users.
In this process, steps 1 and 3 are specific, well-defined and timely actions done by users, whereas steps 2 and 4 happens in a more fluid, processual manner, likely with considerable social interaction to discuss the events from the other steps.
Step 1 is always performed by a single user, steps 2 and 4 require the participation of multiple users and step 3 can be performed either by one or multiple users.

Incentive model

CiviGo incentivizes users by means of two tokens, the reputation token (REP) and the fixer token (FIX). These tokens are non-transferable and minted on-demand following proven action.
REP is awarded to users from steps 1, 2 and 4, that is, to all who are engaging in discussions related to physical issues in the community, building awareness and consensus on what are the most important issues and the validity of solutions implemented in step 3. In other words, it is a token to reward the political and bureaucratic work required for legitimizing issues and implemented solutions. FIX on the other hand is awarded to those who do the physical, practical work of solving the issue.
Both tokens are non-transferable, as they reflect reputation and merit, which are inherently attached to the individual.
Specifically, users receive tokens when:

- Proponents: receive a small amount in step 2 (1 REP), when their proposals get accepted by the community and a larger amount in step 4, at the end of the process. The amount depends on how much participation there was from supporters and validators from steps 2 and 4.
- Supporters: receive by the end of step 4 an amount relative to how much REP they stake. Non-staking users receive a fix amount of 1 REP.

- Fixers: these users receive FIX, relative to the number of supporters in step 2. The reward is split equally among the fixers in case there are more than one.
- Validators: users from this category who are also supporters, receive according to their specific rule previously described. New users who step into the validation process receive a fix amount of 1 REP.

Token Bounties

Fixers receive a bounty (in FIX tokens) equal to the percentage share of REP staked by supporters to that of the total REP pool in the community (bounties are thus always between 1 and 100 FIX), as defined in the following equation.

$$FIX❑_{fixer} = 100 * (REP❑_{staked} \div REP❑_{total})$$

The REP bounty awarded to proponents is equal to the percentage share of supporting and validating members times the percentage share of staked REP (bounties within 1 and 100 REP), as following:

$$REP❑_{proponent} = 100 * ((REP❑_{staked} \div REP❑_{total}) * ((N❑_{supporters} + N❑_{validators}) \div N))$$

Staking supporters receive the same bounty as the proponents, but split proportionally (to the amount of REP staked) among the supporters, defined as follows:

$$REP❑_{supporter_i} = REP❑_{proponent} * (\Re P_{staked_i} / \Re P_{staked})$$

The normalization in the bounty calculation ensures that the incentives remains stable as the average size of REP staking grows.

Disincentives

This incentive system requires also disincentive, to provide a system of checks and balances to CiviGo's DAO communities. Without it, undesired effects could happen such as spam of proposals, selfish alliance between users and trolling in discussions. The disincentive mechanisms preventing abuse, for each user type, are:

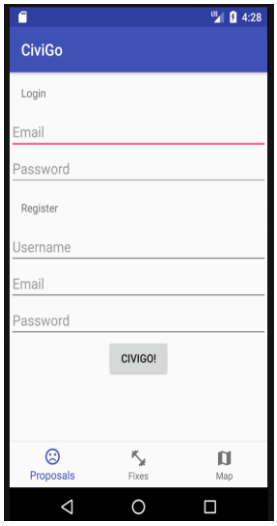
- Proponents: spam is prevented by requiring proponents to stake 2 REP. Additionally they are limited in the number of proposals they may present per week. New users receive 5 REP as they create an account to allow them to make initial stakes.
- Supporters: REP staking is used to disincentive trolling. Supporters staking more REP have more visibility in a discussion, similarly to Reddit. Supporters are awarded in step 4 an amount of REP relative to their staking, in a decaying factor that imposes a hard limit for staking. Supporting is also possible without staking, but a minimum amount of staked REP must be reached for a proposal to be accepted by the community and susceptible to fixing, this minimum is defined as 1% of the total REP pool in the community. A minimum amount of 3 members is also required to approve an initiative.
- Fixers: fixers are already disincentivized by having to do physical work and required to submit evidence, so there is no need for additional measures.

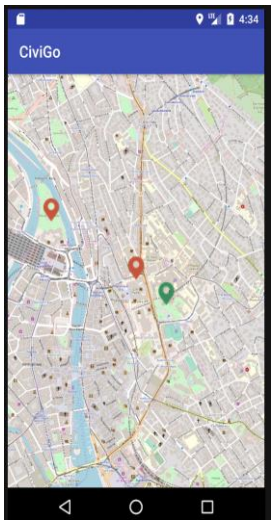
- Validators: for fixes to be validated, a double majority must be achieved, both in numbers of supporters as well as of staked amount. New users may also validate fixes and count as non-staking supporters.

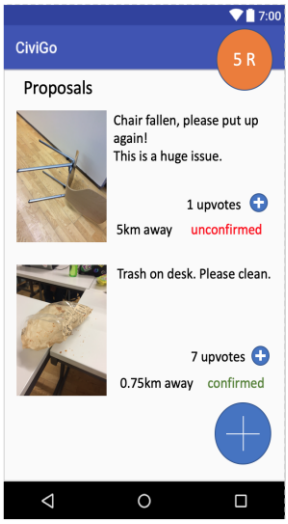
Additional rules exclude fixers from acting also as validators and proponents as acting also as supporters, to avoid conflicts of interest.
The image below illustrates the CiviGo process including the awarding of tokens. The large users and arrows represent the proponent and the fixer, the users who take well-defined action in space, while the online community of supporters and validators is represented in the centre-right and their actions with the red arrows.

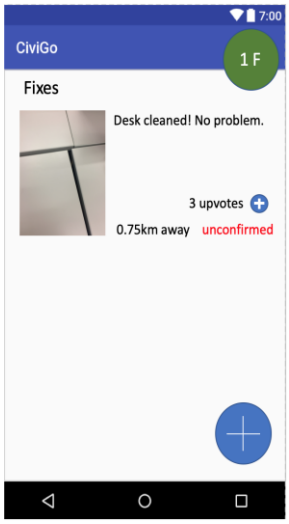
Community interface

CiviGo's interface needs to go beyond simple user interface, as its value lies in the interaction between community members and their environment. For this, proponents have the possibility of printing a small sign containing a QR code together with a message to attract the attention of passers-by to the proposed issue and possibility to engage in the solution. Fixers may print similar signs to engage passers-by to validate their claim to having solved an issue and encourage them to participate in the CiviGo platform.
For the users already participating, a couple of features can turn the experience more engaging. For instance, they may receive push notifications in their smartphones whenever new claims are made in their community. Existing issues are shown in a map, so they may personally visit the location and testify its validity and relevance.
The images below show the conceptual interface of the app displaying these features.









Use case example

The previous figures provide an example use case of the CiviGo process and may be described in more detail as follows.
Let's say someone in the ETH Computational Social Science (COSS, professorship) community spots some messy desk but is in a hurry and can't do anything about it. This person knows about CiviGo and decides to quickly create an account and publish the issue. His initial 5 REP as new member allows him to stake the needed 2 REP to open a proposal, which is submitted by taking a picture of the mess and adding a short description. People from the COSS CiviGo community receive a notification of the new proposal and 7 of them also saw the messy desk and support the issue. 3 of the supporters stake some REP because they are really concerned and immediately comment on the urgency of fixing the issue, while others simply support it without staking since they care about it but not as much. Once the initiative is in the confirmed state, the proponents not only receives their 2 staked REP back but also an initial reward of 1 REP for his work. With the published initiative, people from the community start seeing how much interest is being gathered towards the solution of the issue, represented by the amount of REP staked and number of supporters. Since a total of 5 REP was staked and the current REP pool in the COSS community is 50 REP, there is a 10 FIX tokens bounty for whoever fixes the issue. Once someone with time and interest in helping but also in receiving the FIX tokens fixes the issue, this fixer can solve the issue and then make a claim. People from the community then need to validate the fix, so whoever passes by and sees the issue solved can confirm in the CiviGo app. Whenever a double majority is achieved, in number of validators relative to the total amount of validators and supporters and the amount of REP staked, the bounties are released. The proponent receives 5 REP since 10% of the REP pool was staked and 50% of the community has participated in supporting and validation of the issue. The staking supporters receive 5 REP divided proportionally among them, relative to the amount each one staked. Finally, the non-staking supporters and new validators receive a reward of 1 REP each.

Membership of a CiviGo community

As mentioned before, the CiviGo communities function independently. These communities are public and can be created by anyone. New members can also freely join communities without special permission.
The prevention of membership abuse is achieved also by disincentives. New members may not stake REP as supporters and their support does not count for the double majority if the community has more than 10 members. New members become full members once they reach 10 REP or 5 FIX.
Additionally, the app interface also imposes a limit to membership abuse by requiring the GPS signal to match the location of the issue for fixes and validations.

Literature Review

From a game theoretical standpoint, the previous Finance 4.0 could be modelled as a sequential 2 player game with payouts

A	*Oracle*	Payouts (*A, Oracle*)
claims action (correctly)	grant	1, 1
does not claim action (correctly)	-	0, 0
claims action (wrongly)	deny	x, -1

where *A* is a user of the Finance 4.0 platform who might want to claim an action and *Oracle* is the Oracle who potentially has to grant or deny tokens for such actions.

It is desirable for the Oracle that *A* would only claim actions they committed, but not actions they did not commit (fraudulent behaviour/spamming) since having to deny tokens constantly is work, hence the negative payout in the deny case. What is left to determine is the value *x* to be used. Arguably claiming an action but getting denied the token means *A* wasted effort in even trying to claim the token, so a small negative value should be used. This would solve the whole issue, since that implies *A* never behaves fraudulent. From real life experience it is known however that this is not the case for all possible participants *A* (e.g. internet trolls). There are usually some participants who find value in such behaviour, be it gained exposure, relief of boredom or knowing someone had to make the effort of denying the token. Hence a (possibly small) positive value for *x* has to be considered at least for some participants *A*. This is where the extension proposed by CiviGo comes in.

On first glance some of the mechanisms proposed in the concept section to disincentivize fraudulent behaviour or spamming might seem arbitrary to the reader, but are well founded in game theory, more specifically in signalling theory. The mechanism used is known as *costly signalling* and was first conjectured by Amotz Zahavi in [2]. Although in the original papers only treated the mechanism regarding biological mate selection with textual descriptions, it has since been adopted as a general game theoretical concept and mathematically formalised, e.g. in [3].

The idea is to introduce a cost to a signal (in this case a signal is an action claim) that might be abused for fraudulent purposes such that sending the signal only pays off if it was not fraudulent. In the above model a reasonable cost might be $c = x + (1-x)/2$, in other words a cost halfway between the payout of being fraudulent and correct. Note that this assumes the payoff of fraudulent behaviour is strictly smaller than correct behaviour, i.e. $x < 1$ in this case.

[2] *Amotz Zahavi, Mate selection—A selection for a handicap*

[3] *Colin Camerer, Gifts as Economic Signals and Social Symbols*

The new payoff matrix would then be

A	*Oracle*	Payouts (*A, Oracle*)
claims action (correctly)	grant	1-c, 1
does not claim action (correctly)	-	0, 0
claims action (wrongly)	deny	x-c, -1

Note that $1-c > 0$, i.e. participants still have an incentive to participate correctly, while $x-c < 0$, i.e. fraudulent behaviour is strictly worse that not claiming an action, incentivizing correct behaviour.

In our concept we implemented this cost through staking reputation, the exact amount can be determined by further studying x, in other words by studying how much staked reputation is perceived to be more expensive than a denied claim. The ratios for gained and staked reputation provided in section *Incentive Model* represent an attempt at capturing x in formulae.

Evaluation

A deployment and testing of CiviGo could happen in multiple phases, where an evaluation of the state of the system can be done at each step. Each adding another layer of security and solving some of the open challenges discussed in a later paragraph.

- **Initial Phase:**
  This is a minimal system to attract people to the system and kickstart a community, by providing ease of use.

  The Oracle:
  5 upvotes from different people are required for confirmation.
  The Proofs:
  The only proof is the camera, providing an image of the problem/fix. Basic cheating is detected and prevented by some graph analysis.
  The Blockchain:
  All clients are connected to a central server with email and password, which only logs the activity to the blockchain in whole batches to reduce transaction fees.

- **Improvement Phase:**
  In this phase, the system is further improved, so that cheating is made hard enough and the parameters are adapted, so they scale with the community size.

  The Oracle:
  X upvotes from different people are necessary for confirmation. This parameter could be calculated automatically according to population density in that region. A proposer/confirmer has to stake some of his/her REP to propose/upvote something.
  Improved proofs:
  A proposer/confirmer has to additionally provide GPS location as a proof, that he/she has been there and inspected the legitimacy of the claim. Additionally, the proposer could provide a QR-Code at the problem site, which confirmers can scan to provide further proof (and for ease of use). This could be optional with the additional QR proof leading to a lower required upvote threshold X, as the proofs have more credibility.
  The Blockchain:
  The smart contracts for staking and rewards of the system is deployed. At this point, it may be interesting to enable smart contracts, which award FIX tokens to multiple parties.
  Further tweaks on the amount of the rewards/penalties can still be made.

- **Final Phase:**
  The goal of the project is to get to a stage, where cheating is not worth the expenses and the blockchain gets further decentralized.

  The Oracle:
  The parameter X should be finalized and insights from the prior phase used for this.
  Improved Proofs:
  IoT devices could supersede the old proving mechanisms by using NFC or automatically generated QR-Code.
  Full Blockchain deployment:

The client should also act as a lightweight wallet by this stage. This means we can move another step away from the fin4 centralized servers but would also imply transaction costs for each user.

- **Impact Phase:**
  At this point in the project, we are at a stage where big questions remain, because the platform can get quite powerful with a big community and solid proofs. Any changes at this point cannot be undone easily and have big impact. What this implies and how to try to use it, is a discussion for the next paragraph.

Conclusion and Outlook

Conclusion

CiviGo is a tool to encourage participation in projects of public goods. It uses a social proof mechanism to verify any problems in a 4-stage process. Unconfirmed proposal, confirmed proposal, unconfirmed fix, confirmed fix. Blockchain further improves on portability and proof of reputation (REP) and real-life participation (FIX) within the system.

Challenges

A solved challenge is to confirm an unconfirmed proposal/fix. This is done by the social proof. But the social proof is flawed. To prevent spam and fraud, each upvote or proposal is related with a certain REP staking cost, as explained in a previous paragraph. This means that to be able to gain any REP tokens, one has to first spend some amount of REP. So, if we want people to be able to participate, we have to give them at least a minimal amount of REP tokens when they initially create their account/wallet. This introduces the risk of a Sybil attack, in which a malicious actor could create an arbitrary amount of accounts to upvote his own proposals. This risk is reduced by the proofs which have to be provided alongside a proposal/upvote. In the initial and improvement phase, various existing techniques on the server side can be applied to detect and prevent such behaviour. But in the final phase, where a direct interaction with the blockchain would be planned, we cannot filter out these proposals/upvotes. In that phase, an IoT device would be needed for each proposal for verification. This results in a further challenge about the IoT device where the following unsolved questions arise. Is there a central authority, where these can be obtained? That would destroy a lot of the original idea. Can anyone create such a device? If so, the proof could be faked again. The device also reduces the ease of use drastically for an initial proposer, which is a problem we don’t want to introduce only for being able to go fully onto the blockchain.

Another challenge is the Blockchain transaction cost. In a server-based model, the server operators have to pay for the servers and transactions without any benefit. In the full blockchain model of the final phase, each user has to pay for every action. This would raise the barrier for a lot of people to get engaged with the system.

It thus seems a feasible solution would skip the final phase.

Outlook

CiviGo, a platform to verify real world properties with social proofs, is quite a universal tool. If done right, it could not only be used for small problems, like broken benches and trash, it could be a step towards digitized self-governance. As the platform attracts more people and gets more powerful, sooner or later it starts to get political.

In the far future, the system could be extended with the following ideas:

A proposer could include a budget into the smart contract. As soon as a proposal is confirmed, people can send ETH to that smart contract, which a fixer then can use to realize the project.
The FIX and/or REP tokens can be accepted by external parties for economical/social benefits.

Maybe with CiviGo one day, people will also run around in the city and hunt for real world problems. This could be a realistic scenario, as people are already running around hunting for imaginary monsters.