A Distributed Ledger for Gamification of Pro-Bono Time

Viktor Atterlönn

Högskoleingenjör, Datateknik
2018

Luleå tekniska universitet
Institutionen för system- och rymdteknik
Abstract

Many non-profit organizations such as sports clubs have trouble getting members to help with jobs that benefit the organization in terms of maintaining activities. Within organizations such as these the motive to motivation might not be as clear as in other types of non-profit organizations and often lead to poor involvement from members[14]. The report studies how and if we can improve motivation and involvement in sports clubs using Distributed Ledgers Technology (DLT), Blockchain, and Gamification in the form of a reward and trading system. The result indicates that with such a system it is possible to improve motivation in non-profit organizations such as sports organizations. Gaming elements in the form of a scoreboard were highly appreciated in sports clubs as the majority of the members are contest oriented. The system has its limitations in interacting with events outside the Blockchain but still benefits from the majority of DLT’s features such as high access, high security, and consistent data.
Sammanfattning

Flertal ideella föreningar så som idrottsföreningar har problem med att få medlemmar att hjälpa till med sysslor som gynnar föreningen när det gäller att upprätthålla aktiviteter. Inom organisationer som dessa är motivet till motivation inte alltid lika lätt att finna så som i andra ideella föreningar och leder ofta till dåligt engagemang från medlemmarna[14]. Rapporten studerar hur och om vi kan förbättra motivationen och engagemanget i idrottsföreningar med hjälp utav Distributed Ledgers Teknologi(DLT), Blockchain samt Gamification i form av ett belöning och handels system.

Resultatet indikerar på att det är fullt möjligt att med ett sådant system förbättra motivationen inom ideella föreningar som idrottsföreningar. Gamification element i form utav en resultattavla var väldigt uppskattad inom idrottsföreningar då majoriteten utav medlemmarna är tävlingsinriktade.

Systemet har sina begränsningar vid interagerande med händelser utanför Kedjan men gagnar fortfarande flertalet utav DLT:ns funktioner så som hög åtkomst, hög säkerhet samt konsekvent data.
# Contents

1 Introduction ................................. 5  
1.1 Problem statement ........................................ 6  
1.2 Goals ................................................. 6  
1.3 Scope ............................................... 7  
1.4 Purpose .............................................. 7  
1.5 Thesis Outline ....................................... 7  

2 Background ........................................ 8  
2.1 Distributed ledger technologies .................................. 8  
2.2 Blockchain ........................................... 10  
2.2.1 Blockchain Implementations .................................. 10  
2.2.2 Proof-of-work ...................................... 12  
2.2.3 Proof-of-stake ...................................... 12  
2.2.4 Delegated Byzantine Fault Tolerance ..................... 12  
2.2.5 Public Blockchain .................................... 12  
2.2.6 Private Blockchain .................................... 12  
2.2.7 Permissioned and Permissionless Blockchain ...... 13  
2.3 Smart Contracts ..................................... 14  
2.3.1 Solidity ........................................... 15  
2.4 Gamification ......................................... 16  

3 Method ............................................ 17  
3.1 Selection of ledger ....................................... 17  
3.1.1 Public or private? .................................... 18  
3.1.2 Candidates ......................................... 18  
3.2 Architecture .......................................... 20  
3.3 Detailed Architecture ..................................... 20  
3.3.1 Smart Contracts ..................................... 21  
3.3.2 Communication ..................................... 23  
3.3.3 Frontend design .................................... 25  
3.4 Gamification .......................................... 29  
3.4.1 Badges ............................................. 29  
3.4.2 Level ............................................... 30  
3.4.3 Scoreboard ........................................ 30  
3.4.4 Score ............................................... 30  
3.5 Result & Evaluation ..................................... 31  

4 Discussion ........................................ 33  
4.1 Alternatives to Blockchain ................................... 33  
4.2 Limitations ......................................... 33  
4.3 Future research ........................................ 33  

5 Conclusion ....................................... 35  

Appendices ......................................... 37  
A Questionary ....................................... 37
1 Introduction

In today’s society we almost exclusively rely on financial institutions serving as a trusted third party to process everything from electronic payments to agreements between different participants. This trust based model still works well but it has its weaknesses, there is room for fraud as well as the trusted third party leaking personal information about customers as a result of a cyber attack. What Blockchain and Distributed Ledgers was initially implemented to solve was to remove the trust-based system within electronic payments and instead use cryptographic proof over a peer-to-peer network[1]. The first implementation of Blockchain was a crypto currency called Bitcoin which solved the payment problems between customer and merchant, but what about transactions that contain physical objects or agreements in form of contracts?

Games have always been a fun and nice way of relaxing and doing something not related to work. Lately, we have been seeing more and more gamified elements being introduced into apps used daily. This technique gives the user motivation to keep using the specific application by making the experience more fun and rewarding. This specific technique is called Gamification and has been an extremely hot topic the last couple of years.

Most sports clubs struggle with getting people to put in the work needed for the organization to stay alive[14]. Since the non-profit organizations don’t pay their employees or members for their work they need to be motivated in some other way.

From the beginning, nonprofit organizations have relied on motivation generated from the cause of the organization. This might be enough in organizations which focus on helping other people in need or contribution to a better environment, but how about other organizations such as sports clubs?

Sports clubs are also a form of nonprofit organization but with another goal or cause. Let’s have a kids sports club in mind, they rely on the parents running the organization, but not all parents have the time or motivation to prioritize the organization which leads to problems maintaining activities.

As these two technologies continue growing in popularity the interest in investigating further more about these grow. It will be interesting to see if the collaboration between Distributed Ledgers and Gamification could solve the problem of poor involvement within nonprofit organizations. This thesis test the possibilities of how distributed ledger technologies with the help of gamification can contribute to increasing the work motivation within non-profit organizations such as sports clubs. The test is done by implementation of a distributed reward system specially made for sports clubs and is described and documented later on in this thesis.
1.1 Problem statement

This report studies two different technologies and how these two together could motivate members of nonprofit organizations to contribute more to the organization. Gamification elements working as the motivating part of the system while the Distributed Ledger Technologies focusing on safety of the data and coming to a mutual agreement between users without a trusted third party.

Therefore, the problems statement is as follows:

*Is the utilization of a distributed ledger together with gamification methods a viable solution to motivate people to contribute with pro bono time, for example in a sports club?*

1.2 Goals

In order to fulfil the thesis statement a number of goals needs to be fulfilled. Below one can see each goal and subproblem that needs to be solved. These goals can also be used later to evaluate the result.

Goals:

1. Select gamification methods suitable for members of a sports organization
   (a) Each method should be reasonable to implement in a digital system
   (b) Each method should be motivational or competitive
2. Select rewards that should be available
   (a) Rewards should be relatively easy to obtain
   (b) Rewards should be relevant to the organization
3. Select type of ledger suitable for the system
   (a) Support deployment of Smart contracts
4. Implement each gamification method within the distributed ledger
5. Implement a rewards system within the distributed ledger
6. Design and implement web interface
   (a) The interface should be easy to use
   (b) Provide quick access to commonly used functions
   (c) The interface should be usable wherever the user is located
7. Find a way to allow the interface to communicate with the distributed ledger
   (a) Traffic between interface and ledger should be secure
   (b) Should not care about device or operating system, communication should be possible from Android applications, iOS applications, websites, computer softwares
8. Collect evaluation from a selected sports organization
1.3 Scope

The study covers Distributed ledgers technology as well as how to increase the motivation of pro bono work within non-profit organizations using gamification and other means. Distributed ledger technology serves as the fundamental part of this implementation since it is the part which replaces the trusted third party and keeping the system safe, consistent and trusted. Apart from the technology making this possible the members and users needs to be motivated to use the system, this is done by a method called gamification. By gamifying the system each task will feel more like a game but also contribute to activation of the brains reward system which leads to the feeling of satisfaction.

This study is limited to small tests and user evaluations, the study will not contain an implementation of our solution within bigger organizations and will only serve as an initial design and implementation of a future solution. As this system won’t be tested in a real organization it might need some changes or tweaks in order to fit its purpose.

1.4 Purpose

The potential implementation of Blockchain technology in our daily assignments could revolutionize the way we make trades. The use of Blockchain would remove the factor of needing a trusted third party and thereby contribute to easier trades as well as increased security.

The purpose of this thesis is to implement a distributed trading and reward system with the help of Distributed Ledger Technologies and Gamification. This system will later be used to test the impact it could have in motivating members of nonprofit organizations such as sports clubs. Maybe these two technologies could solve this problem mentioned most non-profit organizations struggle with and thus push the use of Distributed ledgers as well as Gamification.

1.5 Thesis Outline

With the support from Jacob Permansson who I have been working in parallel with I will describe in this thesis how we can implement a distributed reward system for non-profit organizations using Distributed ledgers and Gamification. This thesis is divided into a total of 5 sections where the following section serves as an introduction for the reader about the technology used and studied in this thesis. The third section contains the methodology of building a distributed reward system using distributed ledgers and gamification as well as documentation of some user evaluations. The results are presented in section 3 and will be discussed in section 4 as well as suggestions for future work. At last, I will come to a conclusion which will be presented in section 5.
2 Background

In order to understand everything mentioned in this thesis the reader needs to get familiar with the technology and basics used. Therefore this section starts with an introduction of Distributed Ledgers, how it works and what its used for. Furthermore, the section provides an explanation of Blockchain which is an implementation of Distributed Ledgers. To fully grasp the concept of Blockchain this section will introduce the reader to some fundamental methods used in Blockchains as well as two well-known Blockchains. Finally, a technology called Smart Contracts will be described.

2.1 Distributed ledger technologies

Distributed ledger technologies (DLTs) is a form of a peer-to-peer network where the ledger is replicated over all participating peers which are spread across multiple sites, countries or institutions. The chain usually consist of replicated, shared and synchronized data and with DLTs there is no need for a trusted third party to validate, process or authenticate transactions.

Each participant can access and own a copy of the ledger and if any changes or additions are made to the ledger a new copy is distributed to each participant in a matter of seconds.

Centralized ledgers are more vulnerable to cyber-attacks while DLTs are way harder to manipulate by attacking them. In order for an attack to succeed every distributed copy of the ledger needs to be attacked simultaneously and even if this is possible the DLT is resistant to malicious changes by a single person or party.

<table>
<thead>
<tr>
<th>Centralized</th>
<th>Decentralized</th>
<th>Distributed Ledgers</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Centralized Network" /></td>
<td><img src="image" alt="Decentralized Network" /></td>
<td><img src="image" alt="Distributed Ledger Network" /></td>
</tr>
</tbody>
</table>

- Users (●) are anonymous
- Each user has a copy of the ledger and participates in confirming transactions independently
- Users (●) are not anonymous
- Permission is required for users to have a copy of the ledger and participate in confirming transactions

Figure 1: Two network types and two types of ledgers. [3]
2.1 Distributed ledger technologies

Figure 1 illustrates two types of networks and two types of distributed ledgers. To the left we have the centralized network where all data is stored centrally and to the right of the centralized network we have the decentralized network where the data is spread over several nodes in the network. The two ledgers on the right are two different types of distributed ledgers. The red one is a public ledger where every peer or user is anonymous and can participate in confirming transactions by something called mining or stake. While the blue one is a private ledger where the peers or users can not be anonymous and in order to participate in confirming transactions the user must have permission.

Both these types of ledgers have increased security in comparison to the centralized and decentralized ledgers since these ledgers usually are fully decentralized, meaning that each node has a copy of the ledger.

The public ledger would be the safest of these due to the number of validators that can contribute to the consensus process.
2.2 Blockchain

The first Blockchain was implemented by a person or group under the pseudonym of Satoshi Nakamoto in 2008/2009 to create the well know cryptocurrency Bitcoin. The Blockchain is simply an implementation of a Distributed Ledger, meaning that all data stored on the Blockchain is spread across multiple sites or nodes. The Blockchain is therefore a distributed ledger of economic transactions that can be programmed to record everything of value, such as financial transactions or agreements in form of contracts. [1]

All transactions are stored on a ledger and ordered in time, this ledger is later replicated over all nodes connected to the Blockchain. A transaction is accepted if it’s a valid transaction and is after that grouped into a block which is stored on the Blockchain. For this block to be stored it needs to get an ID, this ID can be found in different ways but usually with the help of something called proof-of-work. This term and another term called proof-of-stake is something that’s commonly mentioned when talking about validation of a block.

Figure 2: Illustration of a blockchain

Figure 2 illustrates a blockchain. The term Blockchain comes from transactions being grouped in blocks or chunks and then connected to a chain of other validated blocks. In order to attach a block to the existing chain of validated blocks one needs to calculate the id of this new block (which is mentioned earlier) which can only be calculated if one knows the id of the last or previous block in the existing chain.

The data stored on the Blockchain can be accessed by proving your ownership which is done by asymmetric key cryptography. The asymmetric cryptography consisting of a private key, public key and an address connected to these keys.

2.2.1 Blockchain Implementations

Before introducing you to different types of consensus processes such as proof-of-work it can be good to know some about different Blockchains that use these different types of consensus methods.

Probably the most know Blockchain and cryptocurrency is Bitcoin. Bitcoin who was the first of its kind allowing payments directly between users without a trusted third party which gave the world a taste of what to come. This Blockchain is a peer-to-peer network which uses the proof-of-work consensus to validate transactions between users and working as a fully decentralized ledger.
As Bitcoin grew in popularity other companies found the interest of developing their own Blockchains, henceforth comes Ethereum.

Ethereum similar to Bitcoin has their own crypto currency called Ether or Gas. But what makes these different is the fact that Ethereum uses the proof-of-stake consensus method as well as something called Smart Contracts.

Ethereum introduced the Blockchain world to Smart Contracts which were completely new, the Ethereum project was not only a cryptocurrency Blockchain but an engine for applications which could be run without a need of a trusted third party. These Smart Contracts are developed in a language called Solidity which was specifically made for this purpose.

Lately, a new up and coming Blockchain have been talked about a lot, NEO. This Blockchain was founded 2014 in China with the name Antshares but later changed the name to NEO. What makes this Blockchain special is that they have taken the Smart Contracts to a new level. As mentioned before Ethereum Smart Contracts have to be developed in Solidity while Smart Contracts on the NEO Blockchain can be developed in some of the advanced languages such as javascript, C#, Java and many more, you no longer have to learn a unique language for development of Smart Contracts.[5]

NEO also uses another type of consensus method called Delegated Byzantine Fault Tolerance (dBFT). NEO’s ultimate goal for their Smart economy is to seamless integrate the traditional economy and digital economy, allowing free flow of all assets.
2.2 Proof-of-work

Proof-of-work is what Bitcoin uses and is the most popular out of those two. In this step, the node is calculating advanced mathematical problems with the goal of finding the ID for the next block of the Blockchain. This calculation needs the ID of the previous block to calculate the ID of the next block, and this makes the Blockchain immutable since if someone wants to change a block you need to calculate the ID of every previous block. Each node participating in the proof-of-work consensus competes in solving the mathematical problem. The first node to solve the problem for the block will be rewarded.

2.2.3 Proof-of-stake

Proof-of-stake in other hands uses a very different approach when validation transactions. A big problem that came with proof-of-work is that every transaction requires energy to be consumed which leads to a very high energy consuming system. This problem were discussed on Bitcoin forums as early as 2011 and in October Proof-of-stake was implemented. Instead of calculating mathematical problems as in proof-of-work to generate new blocks, an owner of the next block would be chosen in a deterministic way where the chance of the person being chosen depends on the wealth (that is stake). This work is now called forging or minting instead of mining as in proof-of-work. In proof-of-stake there is no reward for creating a new block, so instead the minters take the transaction fee. Something that’s worth mentioning is that proof-of-stake is safer than proof-of-work when using a monetary Blockchain.

2.2.4 Delegated Byzantine Fault Tolerance

Is an alternative consensus algorithm to proof-of-work and proof-of-stake. dBFT work to ensure consensus despite illogical behavior by a fraction of participating nodes in the system. In a dBFT there are two types of participants: bookkeeping nodes who take part in the consensus process and then there are normal users. The bookkeeping nodes achieve consensus by voting, two-thirds of the bookkeeping nodes need to approve the transaction in order for a new version of the Blockchain to take place. This protects the Blockchain from radical changes regardless of how much CPU power (Proof-of-work) or coins (Proof-of-stake) an attacker has.[6]

2.2.5 Public Blockchain

A public Blockchain is a Blockchain that anyone in the world can read or send transactions to. Everyone can participate in the consensus process using Proof-of-work or Proof-of-stake. The security of the Blockchain is held intact by crypto economics such as bitcoins. The influence you have in the consensus process is proportional to the amount of crypto economic resources you own [2]. These types of Blockchains are generally fully decentralized.

2.2.6 Private Blockchain

A private Blockchain does not differ too much from a public Blockchain. Instead of allowing everyone to participate in the consensus as on a public Blockchain there is restrictions on a private Blockchain on who that can or can not participate. The nodes that want to participate in the consensus process and validate transactions must have permission from the owner of the Blockchain. Usually, you have a collection of trusted nodes that need to validate each transaction in order for it to go through. The read permission on a private Blockchain can
either be public or private. These types of Blockchains are generally partially decentralized since not all nodes connected to the Blockchain are allowed to own a copy of the data.

2.2.7 Permissioned and Permissionless Blockchain

As a Blockchain can be either public or private it can also be Permissioned or Permissionless. This meaning that each node or user on the Blockchain can either have some type of permission level or all nodes have the same rights. Most permission-based Blockchains are private since these Blockchains are owned by someone that decides who can access the Blockchain, who can participate in the consensus process and who that can own a copy of the ledger. Public Blockchains usually are permissionless since they focus on being fully decentralized and involve as many as possible in the consensus process to ensure high security and consistency of valid data.
2.3 Smart Contracts

The conceptual idea of Smart Contracts dates back almost twenty years from now. Where the thought of Smart Contracts was to embed contractual clauses within hardware and software that we deal with.

"Within a limited amount of potential loss (the amount in the till should be less than the cost of breaching the mechanism), the machine takes in coins, and via a simple mechanism, which makes a freshman computer science problem in design with finite automata, dispense change and product according to the displayed price. A vending machine is a contract with bearer: anybody with coins can participate in an exchange with the vendor. The lockbox and other security mechanisms protect the stored coins and contents from attackers, sufficiently to allow profitable deployment of vending machines in a wide variety of areas." [4]

What we can understand from Nick Szabo is that a Smart Contract is simply a way to enforce trades and agreements without the need of a trusted third party. Due to the locks on the machine an unwanted attacker can not manipulate the machine or take what isn’t his. If we describe Smart Contracts used in Blockchain or on the DLT we can define it as a collection of computer code which can change the state of the Blockchain when executed. These contracts contain a set of predefined functions which can be triggered by sending transactions to the contract. Thanks to the DLT a copy of each contract is stored on every participating peer which makes it immutable and can not be tampered with. [4] Even though the idea of Smart Contracts have been around for a long time there isn’t that many companies known for using them, usage of Smart Contracts are in the early stage. Smart Contracts are used to some extent such as in Bitcoin were a Smart Contract checks if a sender has the tokens he is requesting to send. But the idea of allowing users to deploy their own Smart Contracts into the Blockchain is limited. Ethereum would be the most known company for allowing users to deploy their own Smart Contracts.

Smart Contracts have pros and cons, it’s a great way to ensure transactions and handle payments within a Blockchain. But as soon as you try to integrate the Smart Contracts with the outside world (outside of the Blockchain) it can not ensure the confidentiality of the transactions. The connection between the Smart Contract and the outside world would be an API, and due to this you can not be sure what node the calls come from.

Figure 3: Smart Contract triggered by transactions or events.
2.3 Smart Contracts

2.3.1 Solidity

The knowledge of Solidity is not needed to understand the implementation of the system but is worth mentioning since it is after all used to implement the Smart Contracts. Solidity is an contract-oriented programming language designed to develop Smart Contracts for Blockchains. This language was implemented by the solidity team at Ethereum for the EVM (Ethereum virtual machine)[10]. Solidity is used as a primary language on the Ethereum but is also used on other private Blockchains such as Hyperledger[10]. Solidity is very similar to javascript syntax wise and is therefore easy to learn if familiar with the programming style of javascript. Due to Solidity being relatively new it is constrained to some extent in terms of data types and size of data transactions between Smart Contracts.

Example of creating an object in Solidity.

```solidity
Task memory task;
task.id = numberOfTasks;
task.owner = owner;
task.name = name;
task.valuePerHour = valuePerHour;
tasks.push(task);

numberOfTasks += 1;

return(task.id, task.name, int(task.valuePerHour));
```


2.4 Gamification

Gamification has been a trending topic the last couple of years. The idea of Gamification is to increase user motivation and engagement, such as increased user activity, social interaction or quality put into work. So what is gamification? Gamification is as simple as integrating typical elements of gameplay into other areas of activity that is, another context than games. There are several things that contribute to increased motivation or to capture one’s interest, some of them would be Accomplishment, Social influence, and Ownership. Accomplishment is usually connected to a reward when finishing an action while Social influence is more like bragging to others about your accomplishments and ownership is similar to accomplishment but instead of receiving something you strive to keep what you have accomplished.

The trending topic of gamification has not only been reflected in usage within companies but also in the context of academic work. The number of academic papers published with the focus on gamification have grown rapidly. This suggests that this topic is an interesting topic to study.

Gamification might be a new term and a trending topic right now, but the idea of using game mechanics to solve problems and engage users isn’t as new as you might think. The use of Gamification dates back at least a couple of hundreds of years from now, as the military have been using gamification and simulations, since then they have always been a pioneer in using games and simulations.[7]

Researchers have found that the human brain is hardwired to play, they have also found that there is a connection between learning and playing. Playing leads to faster learning which leads to the human getting smarter. Due to this, it isn’t that strange that Gamification is a trending topic and something a lot of companies have started to implement into their own systems. [7]

Gamification is all about engaging the user, whatever it is the user should be engaged with. But how do you engage the user? Something that’s frequently used when designing games is the MDA Framework which stands for Mechanics, Dynamics and Aesthetics. Mechanics make up for the functionality of the components of the game. These components role is to guide player actions, different components will have different actions. Dynamics is similar to Mechanics but instead of guiding the player the dynamics are the way a player interacts with the game mechanics. Simply, how does the player respond to the mechanics of the game. Finally, Aesthetics is how the game makes the player feel during the engagement and interaction. Aesthetics is simply the outcome of Mechanics and Dynamics and how these two created emotions. [7]

Together these three are implemented into gamified elements such as Points, levels, scoreboards and badges which will be described how to be used and how this thesis uses them in section Method.
3 Method

In this section I explain the method used for this thesis. Initially, I explain what ledger I chose and why. Next, I describe the implementation steps and architecture of the system. Finally, the problem statement is tested and evaluated by some user evaluations.

3.1 Selection of ledger

Before we can start implementing our system we need to decide of what ledger we are gonna use. Since there are several types of ledgers and companies developing solutions for this we need to study these candidates more and decide due to their pros and cons what solution that will suit our system the best. Below I will list some of the candidating implementations of ledgers and describe why I chose the one I did.
3.1 Selection of ledger

<table>
<thead>
<tr>
<th>Blockchain</th>
<th>Public or Private</th>
<th>Permissionless or Permissioned</th>
<th>Smart Contracts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethereum</td>
<td>Public</td>
<td>Permissionless</td>
<td>yes</td>
</tr>
<tr>
<td>Burrow</td>
<td>Private</td>
<td>Permissioned</td>
<td>yes</td>
</tr>
<tr>
<td>OpenChain</td>
<td>Private</td>
<td>Permissioned</td>
<td>yes</td>
</tr>
<tr>
<td>Bitcoin</td>
<td>Public</td>
<td>Permissionless</td>
<td>No</td>
</tr>
<tr>
<td>Ripple</td>
<td>Public</td>
<td>Permissioned</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1: Blockchain selection

3.1.1 Public or private?
The first question would be, should we use a private or public ledger? When answering this question we can narrow down the amount of candidates. Why would we need a public ledger for our system? As described in the background a public ledger would be wanted if we want a fully decentralized ledger where security is our main goal. But since we are building an initial design and implementation of a future system we do not need a public Blockchain. A public Blockchain would slow down the consensus step but also the implementation. Also, this is an encapsulated system, we do not want anonymous users since this will be a trading platform within non-profit organizations.

A private Blockchain will suit our needs better since it will allow us to only give access to wanted users, give permission to trusted users to validate transactions. With a private Blockchain we can increase our working speed by initially creating a ledger with only one validator (that is us) and thereby easily manage the ledger.

3.1.2 Candidates
After answering the question of using a public or private ledger we have come to these two candidates. OpenChain and Hyperledger Burrow. Both are well-known implementations and are private ledgers. Burrow is built upon Ethereum while OpenChain is built from scratch with different technologies. Let’s start describing Burrow then OpenChain, and after that compare these with each other.

Hyperledger Burrow  Built by the team at Monax upon the Ethereum Blockchain. Burrow is a permission Blockchain node constructed out of three components: Consensus engine, EVM (Ethereum virtual machine) and the RPC gateway (API gateway). The ”Permissioned” part means that each node connected to the Blockchain has limitations in what the node is allowed to do. Different nodes can have different permissions, such as validation, reading, sending. The EVM is a well developed VM for Smart Contracts and the consensus process in Burrow follows the Proof-of-stake process which suits the private Blockchain well. Also, Ethereum was the Blockchain that introduced Smart Contracts to Blockchain which means that they have the most experience and probably is the most stable.

OpenChain  Instead of being forked from another Blockchain the OpenChain is built from scratch. Openchain does not use proof-of-stake nor proof-of-work. Instead it’s fully up to the admin of the assets to confirm the transactions. This leads to Openchain being somewhat decentralized, not as decentralized as the Burrow Blockchain. Openchain instead focus on increased speed when working with transactions, which is made possible due to the validation process but also that Openchain does not use the normal
architecture of putting transactions into a block and then store them on the Blockchain. Instead, openchain handles one transaction at the time and validating them instantly.

**New vs trusted** Openchain focus on making the transaction process faster than other Blockchains, this is to suit specific systems that can not allow high latency. The system this thesis document does not need incredibly high transaction speeds and for this reason this functionality is not decisive.

Openchain is still very new and has a new way of thinking regarding distributed ledgers while Burrow is built upon one of the most known Blockchains and follows the proof-of-stake consensus.

After evaluating both Blockchains on the pros and cons I brought up I have come to the conclusion of using Burrow. This is because Burrow is a permission based Blockchain, it uses the proof-of-stake consensus process which we prefer for a private Blockchain. Burrow is also built upon the Ethereum Blockchain which is known for its good VM that handles Smart Contracts excellent. When developing Smart Contracts on the Burrow Blockchain we can use the language Solidity which is very similar to javascript.

The use of Burrow in this project also makes it possible for future work to implement our solution on their Blockchains as long as it's built upon Ethereum which most Smart Contract Blockchains are today. The usage of Openchain would limit this project in a way of making it harder for future work.
3.2 Architecture

The back-end of the system is built upon a block-chain called Burrow with Smart Contracts handling function calls and agreements. The back-end has four contracts handling different parts of the system and one main contract working as a permission layer. All function calls go through the main contract which keeps track of what permission level each user has. The other four contracts work as controllers for their specific area as well as a database.

The contracts is the logic of our application while the Blockchain is the host of our contracts. The Blockchain gives us the possibility to store data and handle transactions in a very secure manner.

Besides the Blockchain and the Smart contracts working as the back-end there is a RESTful API developed with Nodejs which connects the front-end apps with the back-end logic. This API takes requests in form of JSON and then sends a request to the specific contract. When a result is received from the contract the API sends a JSON response to the requester.

The front-end is divided into two areas, one web application, and one android application. The web application is a simple website built with HTML, CSS, and javascript which sends and receives JSON objects from the RESTful API and display the data in a user-friendly way.

![Figure 4: System architecture.](image)

3.3 Detailed Architecture

As described earlier the back-end is built on a Blockchain called burrow which is developed by the team at Monax. This is a permission-based Blockchain forked of the Ethereum Blockchain. The Blockchain has 1 validator account, 1 root account and 20 participant account. These three account types have different permission levels. The validator account is only allowed to read and validate transactions, the root account are allowed to make changes to the Blockchain but not validate transactions and the participant account are allowed to read and send transactions to the Blockchain as well as own a copy of the ledger.

Usually, you would want several validators as this increase the security but in this thesis a more simple Blockchain is used to test the system as this increase the workflow. If wanted the number of validators can be increased easily as long as the number of validators are 2/3 of the accounts registered on the Blockchain.
3.3 Detailed Architecture

3.3.1 Smart Contracts

<table>
<thead>
<tr>
<th>Contract</th>
<th>Functionality</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>Main contract, working as a permission layer</td>
<td>No</td>
</tr>
<tr>
<td>Users</td>
<td>Handles everything regarding users as well as storing user data</td>
<td>Yes</td>
</tr>
<tr>
<td>Tasks</td>
<td>Handles everything regarding Tasks as well as storing task data</td>
<td>Yes</td>
</tr>
<tr>
<td>Badges</td>
<td>Handles everything regarding Badges as well as storing badge data</td>
<td>Yes</td>
</tr>
<tr>
<td>Points</td>
<td>Working as a bank, keeps track of points owned by every user</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 2: Smart Contracts on the Blockchain

On the Blockchain there are Smart Contracts (table 2) working as controllers as well as databases. The first contract called Organization works as a permission layer or gate. This contract keeps track of each user's permission level as well as the addresses to each controller and database contract. Function calls that need certain permissions are sent through this contract before they reach the controller contract, this is to deny non-permissioned users to tamper with the system.

The points contract works as a bank, it handles the points for each user. Points are received when completing a task and can only be spent within the system on either rewards or services from other users or organizations. This contract makes sure that the user has enough points to make the desired trade as well as giving the correct amount when completing a task.

The user’s contract handles everything regarding the user, it stores user relevant information such as name, email, badges earned by the user. This information is limited in the term of visibility for other users. This user specific information is mainly used in the confirmation process of a task and will only be visible for the organization admin.

Registration of a new user also goes through this contract as well as removing already existing users.

Both the badges and tasks contracts take care of managing existing badges and tasks as well as creating new ones.
pragma solidity ^0.4.4;

contract Badges{
    address public owner;

    uint numberOfBadges;

    struct Badge{
        uint id;
        bytes32 name;
        bytes desc;
    }

    Badge[] public badges;

    function Badges(){
        owner = msg.sender;
        numberOfBadges = 8;
    }

    function createBadge(address sender, bytes32 name, bytes desc) returns(uint, bytes32, bytes){
        if (sender != owner) return;

        Badge memory badge;
        badge.id = numberOfBadges;
        badge.name = name;
        badge.desc = desc;
        badges.push(badge);

        numberOfBadges += 1;

        return (badge.id, badge.name, badge.desc);
    }

    function deleteBadge(address sender, uint id) returns(uint, bytes32, bytes){
        if(sender != owner) return;

        var index = findBadge(id);

        var badge = badges[index];

        for (uint i = index; i < badges.length-1; i++){
            badges[i] = badges[i+1];
        }

delte badges[badges.length-1];
badges.length--;

        return(badge.id, badge.name, badge.desc);
    }
}

Figure 5: Badge contract.
Figure 5 shows a draft from the badge contract. In the top of the file there is a line telling the computer that this is a solidity contract. This line is something that is required in solidity and will be recurring through all contracts on the Blockchain. Below this line the Contract is declared with a name as well as functions, structs and variables. The structs is a declaration of an object that can be used to store several variables which in this case will store a badge id, name and description. This contract will have an array where all available badges will be stores as well as a constructor which states who is the owner of the contract (admin) and initialize a counter variable to keep track of how many badges that are available.

The create function looks the same as the code part introduced in section 2.3.1, simply create a badge object and push it to the array.

The delete function looks differently from the create function. First the function checks that the caller is permissioned to do this action. If so is the case, iterate through the badge array and find the queried badge and then remove it.

### 3.3.2 Communication

<table>
<thead>
<tr>
<th>Route</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>get /users</td>
<td>Route to get all users</td>
</tr>
<tr>
<td>post /signin</td>
<td>Route to sign in to the system</td>
</tr>
<tr>
<td>/user/register</td>
<td>Route to register a new user</td>
</tr>
<tr>
<td>/user/changepassword</td>
<td>Route to change signin password for a user</td>
</tr>
<tr>
<td>/user/get</td>
<td>Route to get a specific user</td>
</tr>
<tr>
<td>/user/getbyaddress</td>
<td>Route to get a user by address</td>
</tr>
<tr>
<td>/user/remove</td>
<td>Route to remove a user</td>
</tr>
<tr>
<td>/user/givebadge</td>
<td>Route to give a badge to a user</td>
</tr>
<tr>
<td>/user/registertask</td>
<td>Route to register a task as a user</td>
</tr>
<tr>
<td>/user/tasks</td>
<td>Route to get all tasks done by a user</td>
</tr>
<tr>
<td>/user/badges</td>
<td>Route to get all badges collected by a user</td>
</tr>
<tr>
<td>/task/create</td>
<td>Route to create a new task</td>
</tr>
<tr>
<td>/task/remove</td>
<td>Route to remove a task</td>
</tr>
<tr>
<td>/task/updatevalueperhour</td>
<td>Route to change valueperhour of a task</td>
</tr>
<tr>
<td>/task/updatename</td>
<td>Route to change name of a task</td>
</tr>
<tr>
<td>/task/get</td>
<td>Route to get a specific task</td>
</tr>
<tr>
<td>get /tasks</td>
<td>Route to list all available tasks</td>
</tr>
<tr>
<td>/point/send</td>
<td>Route to give points to a user</td>
</tr>
<tr>
<td>/point/get</td>
<td>Route to get points a user have earned</td>
</tr>
<tr>
<td>/scoreboard</td>
<td>Route to get the current scoreboard</td>
</tr>
<tr>
<td>/badge/create</td>
<td>Route to create a new badge</td>
</tr>
<tr>
<td>/badge/remove</td>
<td>Route to remove a badge</td>
</tr>
<tr>
<td>/badge/updatename</td>
<td>Route to update the name of a badge</td>
</tr>
<tr>
<td>/badge/updatedesc</td>
<td>Route to update the description of a badge</td>
</tr>
<tr>
<td>/badges</td>
<td>Route to list all available badges</td>
</tr>
</tbody>
</table>

Table 3: All API routes
As mentioned earlier the interaction between the Smart Contracts and the frontend of the application is done through an API developed in NodeJS, the API calls are accessed through HTTP-requests. In the table above (table 3) all possible interactions are listed in form of API-routes. Some of these routes need admin permission as they alter the settings regarding tasks, badges and users.

![Figure 6: Flowchart of Signin process](image)

Figure 6 shows the process when a user signin via the frontend application. There are more scenarios then only signin but its a great example to show how the flow of the system works since the flow is similar for every action available in the frontend application. The user interact with the frontend element, data is sent to the API which sends a request to a specific Smart Contract on the chain, the Smart Contract executes a function which sends back a response to the API, this response formatted correctly at the API and then sent back to the frontend application which renders something new depending on the respons.

In the scenario presented in figure 6 a user is shown a signin page with an email and password field as well as a signin button. The user enter the credentials and press the signin button. When the button is pressed the value of the email and password field is sent to correct API-route ('/signin'). When the API recieves the credentials it sends a request to the User contract which will search for a user with this email and password combination. If a user is found the User contract sends back a user object containing details about the user as well as an address specific for this user. If no user is found the User contract sends an error back to the API. The API recieves the respons from the User contract and forwards it to the frontend. If the respons is an error, the user will return to the signin page, prompted to enter the correct credentials. But if the respons is an User object the user is redirected to the user dashboard.
3.3.3 Frontend design

Figure 7: User dashboard. When signed in the user have the possibility to register a task, view completed tasks, view number of badges collected, view points collected as well as viewing the current level.
3.3 Detailed Architecture

Figure 8: Page where a user can see the current scoreboard, points collected within the organization as well as the name of the organization.
Figure 9: Some information about the system and a progressbar to the next level
Figure 10: Information about all currently available badges
3.4 Gamification

In order to motivate the users of the system to help within the organization some elements of gamification was implemented in the system. This section will describe these elements as well as why these where chosen. A lot of inspiration regarding the selection of gamification elements where taken from Emiliens work [2]

3.4.1 Badges

One of the most known elements of gamification is achievements and badges. Both work in a similar way, do something and get rewarded and recognized for the thing you have done. Since both work in a similar way this system only uses badges as these can be presented in a more visually attractive way.

Anyone who is familiar with gaming have heard of the gaming platform "Steam", this platform introduced gamification in form of achievements and badges which gave more content to each game hosted on this platform, not only could the user play through the game as it was implemented to be played but also they could now perform specific task along the way and get recognition for it.

The idea of badges instead of achievements is also a lot more cheap to implement into the system which increases the performance of the back-end. The visual part of the badges is handled by the API server since storage of images on the Blockchain was rather difficult to achieve as well as we want to limit as much of the gamification as possible to the front-end which also contributes to an increase in performance regarding the Blockchain.

<table>
<thead>
<tr>
<th>Badge</th>
<th>Qualification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook</td>
<td>Make a post on Facebook about the organization</td>
</tr>
<tr>
<td>Twitter</td>
<td>Make a post on Twitter about the organization</td>
</tr>
<tr>
<td>Instagram</td>
<td>Make a post on Instagram about the organization</td>
</tr>
<tr>
<td>Friend</td>
<td>Invite a friend</td>
</tr>
<tr>
<td>Friends</td>
<td>Invite 10 friends</td>
</tr>
<tr>
<td>Leader</td>
<td>Achieve rank 1 on the leaderboard</td>
</tr>
<tr>
<td>Competitor</td>
<td>Reach the top 5 on the leaderboard</td>
</tr>
</tbody>
</table>

Table 4: Badges available in the system

Each badge consists of an identifier in term of name and icon and a description which states what has to be done to get the badge. Above you can see the current badges available in the system and as mentioned before the integration of real world events with the Smart Contracts is something that is lacking with Blockchain. Due to this some of these badges have to be given by the organization admin.
3.4.2 Level

A new level is something you reach after collecting a specific amount of points. After reaching a new level the goal is reset and a new goal is set to reach the next level. These levels are a way of displaying how much time a person put into the organization. This in a similar way as badges will contribute to the reward feeling as well as giving the user a way of bragging or showing other users how much time they have put into the organization. These ”level points” are the cryptocurrency you get after completing a task, these can also be spent on other rewards as well as traded for services from other users or organizations.

When consuming cryptocurrency they will be taken from the user but the ”level” will still stay the same as it counts the amount of currency a user has collected over time and not how much it has at the moment.

3.4.3 Scoreboard

The Scoreboard is a way to check what other users have contributed with. The Scoreboard shows the name of the user, email, score or points and level. This is something that contributes to increased motivation in form of competition. The previous gamification methods focus on increasing a users motivation by rewards, while this might work for some people it might not work for all.

Something that very common within sports clubs is that most members are competitive, due to this a scoreboard can revive the competitive spirit and make the user spend more time in the organization.

The idea of a scoreboard will also put the ones not contributing in the spotlight which could lead to them contributing more.

3.4.4 Score

The score is the foundation of most gamification elements in the system. By collecting points or score the user increase in level as well as climbing on the scoreboard. For now the score is generated within an organization and handed out to a member after completing a task within the system. Each task have a value per hour that tells the system how much a member should be given after completing a task that took x amount of hours. Currently this ”value per hour” is a arbitrary value but should later on have a specific value proportional to all tasks available.

The score also have the potential to work as a cryptocurrency and is intended to do so in future system.
3.5 Result & Evaluation

In order to get an understanding if this system could be a possible alternative to increasing the voluntary work within ideal organizations a questionary was sent to a group of people for their opinion.

This group of people was selected from an organization called "Järva MK" who is dependent on parents putting time into the organization to keep it running. Today Järva MK is struggling with this and need to find a solution, thereby making them a perfect candidate.

This form simply asked the user about their previous experience with a computerised system, how much time they contribute to the organization, what keeps them motivated and what they think about this system we built, if they see any potential in it.

<table>
<thead>
<tr>
<th>Question from Appendix A</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100% Male</td>
</tr>
<tr>
<td>2</td>
<td>Between 45-64</td>
</tr>
<tr>
<td>3</td>
<td>50% Use computers regularly, 50% Sometimes use computers</td>
</tr>
<tr>
<td>4</td>
<td>50% Never used gamified systems, 50% Used gamified systems before</td>
</tr>
<tr>
<td>5</td>
<td>Everyone that have used gamified systems indicated that the system increased their motivation</td>
</tr>
<tr>
<td>6</td>
<td>&gt;10 hours/week during a season, around 260 hours/year</td>
</tr>
<tr>
<td>7</td>
<td>&quot;My personal motivation to help&quot;, &quot;Improve, do one's share&quot;</td>
</tr>
<tr>
<td>8</td>
<td>100% would like to use this system if introduced in the organization</td>
</tr>
<tr>
<td>9</td>
<td>100% Thinks that this system could help motivating members of the organization</td>
</tr>
<tr>
<td>10</td>
<td>&quot;Competetive, Rewards, clear expectations&quot;, &quot;Will be somewhat of a competition. Those who don't help within the organization will be portrayed, if the points are visualized.”</td>
</tr>
<tr>
<td>11</td>
<td>n/a</td>
</tr>
<tr>
<td>12</td>
<td>100% Thinks a majority will be honest with the tasks they register as done</td>
</tr>
<tr>
<td>13</td>
<td>100% Thinks there should be some kind of admin that confirm that a task has been done</td>
</tr>
<tr>
<td>14</td>
<td>&quot;Tasks are registered in an application, admin confirm that registered task are done&quot;</td>
</tr>
<tr>
<td>15</td>
<td>&quot;Discounts, Buy services from other organizations&quot;, &quot;Memberfees, VIP-activities, Trade for services from other members or organizations&quot;</td>
</tr>
</tbody>
</table>

Table 5: Data collected through questionary
Tabel 5 shows the result for each question from the questionary. Unfortunately the number of members answering the questionary was few but those who answered gave some interesting answers. Eventough the number of participating members was few the answers are very relevant to the case.

The response was very positive as most participants thought a system like this could have a big impact on their organization. Most of the persons answering the questionary were already putting a lot of time into the organization and what their motivation to do this was mostly the feeling of needing to help. They thought the possibility of collecting points and getting recognized for the work they do was an interesting idea and something they more than gladly would use. Something that also was mentioned in the answers was that the members of the organization are incredibly competitive so the gamification elements and especially the scoreboard would contribute a lot to increase the member’s motivation.

The technology used to implement the system was not brought up in the questionary since its not relevant to the user. The parts of the system that was relevant to the user was the gamification elements. Below the result for each element is presented.

The badges did not get that much attention from the members answering the questionary. The current badges in the system might not be that relevant to the users and would possibly need some more thought to for a better result. From the result of the questionary a conclusion can be made that badges are not needed for this version of the system. They are though worth to keep around since new versions of badges could fit the system better then the ones existing at the moment.

Similar to the badges, the levels did not get that much attention. The levels are though a part of the scoreboard and a result of collected points or score and is worth to keep around.

The scoreboard was the gamification element who got the most attention. Since most members answering the questionary are competetive they thought a scoreboard could contribute to increased motivation. If not increasing motivation, the scoreboard would atleast show which ones who help within the organization and which ones who do not help and thus force members to help.

The score or points have been the main gamification element since all other gamification elements besides badges have some sort of connection to it. Also, similar to the scoreboard, the score got alot of attention from the members. For now the score is only used to increase in levels and climb on the scoreboard but the idea of making the score a currency was highly appreciated among the members of the organization.

To summarize the result of the questionary. This specific organization thinks that a system like this could solve their problems with getting members to spend pro bono time into the organization. Today the leading factor of user motivation was the feeling of needing to help and improve. The use of our systems gamification elements and the possibility of trading collected point between users would give the member both motivation in form of competition and the craving of getting rewards.
4 Discussion

4.1 Alternatives to Blockchain

This system could be developed without the blockchain and still solve the problem within nonprofit organizations. One could for example use a central database such as SQL and build a server with javascript which would handle all the back-end logic. The enduser of the system would not be able to distinguish the difference between the two solutions. But this solution would not get rid of the trusted thirdparty nor would the security of data storage or transaction handling be close to what a Blockchain would offer. The main purpose of this thesis is not only to solve the problem within nonprofit organizations but also examine the technology of Blockchain and Gamification, thus makes the Blockchain the correct choice.

4.2 Limitations

Similar as in other DLT systems this system also has its flaws due to the amount of interaction with the outside world. The system still benefits of the Distributed Ledger but due to the high amount of outside Blockchain events the system could be fooled.

Since the points a user can earn are paid from an organization, fooling the system would mean that the organization would be the one doing it. But since an organization which depends on their reputation have too much at stake, this would be very unlikely to happen.

But if this for some reason would happen we could fix it by predefining the points each task would be worth or we could implement a stage between starting a task and getting paid in form of a new agreement contract which states the task and the amount that should be paid. And when the task is done these points are given to the user.

Another limitation of this system would be, let’s say a person trades some points for a service from another user. This person thinks that the hired person did a bad job and does not want to pay while the person who got hired thinks he did a good job. Who is right and who is wrong? Unfortunately, this problem exists in most systems dealing with trades between non-professional users and would have to be solved in an unbiased way.

Finally, when introducing more organizations into the system there has to be some kind of economic spreadsheet to follow to eliminate the possibility of one organization paying their members more than other organizations would pay for the same task.

4.3 Future research

There are several alternatives to future research within this project. The first thing would be to removing the storage of sensitive user-data on the Blockchain and allowing each user to own their own data. This could be solved with the help of paper wallets which has been mentioned and used by other Blockchains such as Ethereum[2]. This is not due to the Blockchain being insecure but due to the idea of allowing the specific user to own their own data instead of the owner of the Blockchain to own each users data.

When this is solved the idea of broadening the system and allowing more than one organization to exist on the Blockchain would be ideal as members of different organizations could trade their earned points for services from other members. Another idea would be that an organization could hire a member of another organization during specific event in exchange of points.

Lastly, something that was mentioned from members of the organization was to implement different types of rewards such as discount on member fees, VIP-Activities or even introduce
goals that have to be reached in order to not have to pay a penalty.
As you can see the alternatives are many.
5 Conclusion

This thesis studied two new technologies, Distributed Ledger Technologies and Gamification. Later on, with the knowledge of these two implement a distributed trading and reward system to solve the motivation problem within non-profit organizations such as a sports club. So, Is the utilization of a distributed ledger together with gamification methods a viable solution to motivate people to contribute with pro bono time, for example in a sports club?

The final conclusion of this thesis is that a reward system like this can absolutely contribute to the problem stated as well as pushing the use of Distributed Ledgers and Gamification. The leading factor in motivating people in these organizations is Gamification and as in most sports clubs the competitive spirit is high so a gamified element in form of a scoreboard would be ideal for motivation, which was confirmed in the user evaluation. This system focused on a sports club but could be used in most nonprofit organizations, the gamification elements might have to be tweaked in order to suit better.

The system still has its limitations due to the high amount of contact with the "outside world" but this is something that has to be researched more about and hopefully is solved in the near future.
But until then the system still benefits from the Distributed Ledgers advantages as high decentralization, leading to increased security and high availability. These two technologies are super interesting to work with and hopefully, we will see more use of them in the near future.
REFERENCES

References


Appendices

A Questionary
**Belöningssystem för ideell förening**


T.ex Järva MK saknar flaggvarter till helgens tävling. Järva MK lägger då ut en ansökan i systemet vilket öppnar upp för medlemmar i andra föreningar att hjälpa till och tjäna poäng.


* Required

1. **Kön? **
   
   *Mark only one oval.*
   
   - Man
   - Kvinna

2. **Ålder? **
   
   *Mark only one oval.*
   
   - 12-17
   - 18-24
   - 25-34
   - 35-44
   - 45-54
   - 55-64
   - 65 eller äldre

3. **Din erfarenhet av datorer eller datorbaserade system? **
   
   *Mark only one oval.*
   
   - Aldrig använt
   - Använd någon gång
   - Använder ibland
   - Använder ofta

4. **Har du använt något poängbaserat system förut med nivåer och medaljer? **
   
   *Mark only one oval.*
   
   - Ja
   - Nej
5. Om Ja, gav denna spelifiering (poäng, nivåer, medaljer) dig mer motivation till engagemang inom det berörda systemet? 
Mark only one oval.

☐ Ja
☐ Nej

6. Hur mycket tid lägger du ned på frivilligt arbete som gynnar föreningen? *

7. Vad får dig att engagera dig i föreningen? *

8. Skulle du kunna tänka dig att använda detta system om föreningen inför det? *
Mark only one oval.

☐ Ja
☐ Nej

9. Tror du att detta system skulle kunna få dig och andra i föreningen att lägga ner mer tid på föreningen? *
Mark only one oval.

☐ Ja
☐ Nej

10. Om ja, kan du kort beskriva varför?

11. Om nej, kan du kort beskriva varför?
12. Tror du att medlemmar i föreningen kommer vara ärliga med att endast registrera uppgifter som verkligen genomförts? *
   Mark only one oval.
   ☐ Alla
   ☐ En majoritet
   ☐ En minoritet
   ☐ Ingen

13. Bör det finnas någon som bekräftar att uppgiften är genomförd och därefter få poäng? *
   Mark only one oval.
   ☐ Ja
   ☐ Nej

14. Om du svarade Ja på föregående fråga, hur kan man tänkas gå tillväga?

15. Hur skulle du vilja kunna använda insamlad valuta? t.ex förmåner *

16. Har du någon idé på belöningar utöver pengar som skulle få dig och andra medlemmar att lägga ner mer tid på föreningen?