UNIVERSITY OF SÃO PAULO

Institute of Mathematics and Computer Sciences

Department of Computer Systems

BLOCKCHAIN TECHNOLOGY APPLICATIONS FOR FINANCIAL TRANSPARENCY IN NON PROFIT ORGANIZATIONS

IGOR VINICIUS ALVARENGA MARINELLI

São Carlos - SP - Brazil

BLOCKCHAIN TECHNOLOGY APPLICATIONS FOR FINANCIAL TRANSPARENCY IN NON PROFIT ORGANIZATIONS

IGOR VINICIUS ALVARENGA MARINELLI

Advisor: FERNANDO SANTOS OSÓRIO

Monography referring to the course completion project within the scope of the discipline SSC0670 - Graduation Project I of the Department of Computer Systems of the Institute of Mathematical and Computer Sciences - ICMC-USP to obtain the title of Computer Engineer, in collaboration with University of California, Berkeley.

Concentration Area: Blockchain.

USP – São Carlos
03 June 2019

Dedication

For Aleksandra Jarocka, who encouraged me to write my bachelor thesis and drop out right after.

Acknowledgment

To my Professor Fernando Osorio, thank for taking the time to guidance and providing me the inspiration to do this and to go further. I wish everyone could have a Professor that believes students like you believed in me.

Abstract

In 2016, an estimated USD 500 billion was donated to charities worldwide. Charities often rely on third-party marketing firms for their fundraising activities to better compete for a portion of these funds. Traditional marketing methods used by these firms such as telemarketing and street advertisement are failing to find traction with Millennials. In addition, many of these firms have engaged in tactics that undermine the trust of the public and ruin the reputation of the charity industry. In an age where we can track the delivery of a pizza or see how far away our Uber is, why can’t we track our money from the moment we donate all the way to the cause? Millennials expect more transparency and control over their donations, including the ability to connect emotionally with receivers by seeing the direct impact their contribution has made. As the majority of charitable donations come from an ageing population, fundraising approaches need to better engage Millennials or there will be a significant reduction in donations from the general public in the next 10 to 20 years. This project highlights the main issues inhibiting people from giving and demonstrates how to use blockchain technology to radically transform the giving experience. In order for people to see exactly where their money is going and the proof of impact of their action, ensuring each transaction is transparent by using an immutable public ledger. Unique Digital Identities (UDID) are created to validate receivers and initiate smart contracts, ensuring funds are only used when the desired impact is confirmed.

Key-words: Blockchain, Ethereum, Cryptocurrency, Smart Contract, UDID, Fundraising, Crowdfunding, Millennials.

Table of Contents

List of Abbreviations and Acronyms

BCF	Blockchain Charity Foundation
ERC	Ethereum Request for Comment
EVM	Ethereum Virtual Machine
ICO	Initial Coin Offering
MSF	Medecins Sans Frontieres
MVP	Minimum Viable Product
NGO	Non-Governmental Organisation
P2P	Peer-To-Peer
UDID . . .	Unique Digital Identities

List of Tables

List of Figures

List of Source Codes

CHAPTER 1: INTRODUCTION

1.1. Contextualization and Motivation

In this chapter, the problems regarding the world of non profit organisations and will be presented. Charities, NGOs and social enterprises - the organisations at the forefront of solving the world's social and environmental problems - are in a bind. In the next sections we'll dive into the major causes of what's holding the social sector funding back.

1.1.1. State of Fundraising Worldwide

Charities rely on donations from corporations and the general public. While there are no total worldwide figures available showing exactly how much is given each year, it is estimated that in 2016, approximately USD 500 billion was donated to charities by individuals and corporations.* To clarify, Table 1 presents a breakdown of some of the major countries:

Country	Local donations	USD value
Australia	$30bn [1]	$23bn
Brazil	R$15.7bn [2]	$4.2bn
China	¥139bn [3]	$22bn
United Kingdom	£9.7bn [4]	$13.6bn
United States	$390bn [5]	$390bn
Total:		$452.8bn

Table 1 - Personal and corporate donations worldwide.

*As the five countries mentioned in this section account for 452.8bn, this estimate conservatively assumes remaining countries would round total donations up to at least 500bn. Currency chosen is U.S. Dollars to facilitate operations throughout the monograph.

1.1.2. Fundraising Is Expensive

Globally, the number of charities is increasing (with over 1 million public charities in the United States alone [6]), creating more competition for donations each year. At the same time, budget cuts from governments causes a reduction in grants to non-profit organisations.

Each charity not only requires funds to help their associated causes, but also to meet ongoing administration and operational costs. To keep these costs low, charities rely on volunteers who are passionate about the cause but can lack relevant sales skills when it comes to fundraising [6]. Charities tend to outsource marketing and fundraising to intermediaries who typically rely on unsophisticated interruption methods such as cold calling, billboards and street canvassing. Many of these intermediaries don't leverage best practices of social sharing, content and mobile device driven marketing. Reports estimate that up to 83% of donations can be lost to fundraising costs [7].

Outsourcing has had moderate success over the last decade, with the majority of these funds coming from monthly direct debit subscriptions [8], i.e., monthly donations. Therefore, marketing firms are able to charge charities exorbitant upfront fee of between 800% to 1700% of the value of each monthly subscription obtained [8]. To illustrate, if subscriptions are $30 a month, the charity will be charged up to $510 per person. If the marketing firm signs up 1000 people, they will be owed a commission of $510,000 payable upfront.

To give a more concrete example, Medecins Sans Frontieres (MSF) has shown in their financial report of 2018 [Annex B] an income in personal and corporate donations of €1.501bn in 2017 and €1.482bn in 2018, not only there was a decrease from one year to another, but also they follow the same pattern as shown above.

1.1.3. Major Concerns

Lack of Impact: It can take up to 17 months for the charity to break even, so donors wanting to contribute for shorter periods actually cost the charity money rather than having an impact on the cause.

Lack of Access: This high upfront cost makes this marketing option excluding for smaller charities, as they often do not have the necessary capital available.

Traditional fundraising methods have been somewhat profitable for charities in the short term; however studies have revealed that while the total amount given has gone up, it is coming from fewer people [8]. Over the last decade, there's been a sharp decline in the percentage of people donating in response to appeals via telemarketing, street, mail, television and door knocking. Research suggests that the biggest drop has occurred in street fundraising [8].

In 2016, only 19.3% of people who were approached on the street actually followed through with a donation, compared to 65.7% in 2005 [9].

This raises the question - 'What will happen to the long-term sustainability of charities as response rates to traditional charitable fundraising methods continue to decline?'

1.2. Generational Differences

1.2.1. The Millennial Way of Charity

A report from the USA revealed that the Baby Boomer generation (born 1946 to 1964) accounted for 43% of the total amount donated [10], roughly four times that of Millennials (born 1981 to 1995). There was almost 50% more engagement from Boomers (51 million donors) than Millennials (32.8 million donors) [10], suggesting the difference is not simply due to higher disposable income. This indicates that charities and marketing firms are not using methods that appeal to the demands and preferences of younger generations, who have a stronger aversion to intrusive telemarketing and face to face appeals. As Nicholas Fandos, Congressional correspondent for The NY Times, smartly summarized [11]: "Millennials expect transparency, sophisticated storytelling and technical savvy from their charitable organizations. And many donors will not only give money, but will also volunteer and lend the force of their own social networks to a cause they believe in... they want to be able to see and measure how those gifts are making a unique impact."

Nevertheless, Millennials prefer quick and easy donation methods that provide instant gratification and transparency. When we look at direct mail for example, both Millennials and Boomers generally believe that receiving direct mail is acceptable (Millennials 66% and Boomers 61%), yet Boomers are four times more likely to take action and donate in response [10]. Almost two-thirds of Millennials and nearly half of Gen X would prefer to donate using their smartphone, but little is being done to cater for this demand.

When it comes to Gen Y versus Boomers, both believe that receiving direct mail is acceptable (Gen Y 66% and Boomers 61%), yet Boomers are four times more likely to take action and donate in response [11].

Approximately 20% of online donations come from smartphones and tablets – however, the current processes are overly complicated and time consuming, resulting in over 50% of people giving up after attempting it [12]. In addition to this, traditional channels (door to door, street canvassing and telemarketing) are now being described by the majority of people as unacceptable forms of charitable fundraising [10].

1.2.2. Uncomfortable Marketing Tactics

By outsourcing their fundraising activities, some charities seem to benefit well in the short term. However, they're prioritising this short-term gain over a long-term positive relationship with the general public by turning a blind eye to marketing tactics that can border on bullying. Over a third of people (35%) rate their experience with charity fundraisers as "bad". The most common issues reported were that they felt harassed, pressured into giving more than they can afford and general disappointment in how the sector is being run [13].

People are also tired of being interrupted by unsolicited phone calls while going about their day and being made to feel guilty for saying "no" to fundraisers. This experience actually creates an aversion to giving – with 66% of people indicating they are less likely to give to charities that do this [8].

1.2.3. Public Trust Issues

Around 1/3 of people don't trust charities [14]. This is because the non-profit fundraising industry has been plagued with uncomfortable marketing tactics, stories of unethical treatment of staff and reports of minuscule amounts from donations actually making it to the advertised recipients.

It's little wonder people are now demanding more transparency and accountability from charities. Nearly 60% of Millennials and 50% of Gen X say that seeing the direct impact of their donation will influence their decision to give [10] – and yet the industry is struggling to find solutions to meet this demand.

Most charities do not have a system of data sharing that lets each other know who they are helping and how [13]. Unfortunately, this can lead to situations where some recipients exploit the system by registering with multiple charities – at the expense of those who have genuine needs. This problem provides some politicians and the media with a biased justification to discriminate against welfare recipients collectively, unfairly targeting the most vulnerable people in society and reducing the public's desire to give.

1.2.4. Crowdfunding

Crowdfunding websites showcase and provide financial transactions for nonprofits individuals or groups attempting to get a large number of people to donate small amounts of money for their project or cause. Crowdfunding can be an alternative or a complement to traditional financial circuits that NGO's have been through.

Therefore, crowdfunding platforms rarely specify in detail their own obligations after funds have been collected. Large donors are perhaps more likely to follow up to ensure that donated funds are well spent, and there has been massive scams related to crowdfunding worldwide [15].

1.3. Problem Summary

Expensive Intermediary Costs: Third-party fundraisers, international money transfers and internal charity staffing costs can take up the majority of funds raised by a charitable organisation, leaving only cents from each dollar to make an impact on the actual cause.

Outdated Fundraising Methods: Younger generations are opposed to cold calling, door knocking and other methods adopted by fundraising intermediaries. They would rather be proactive and give on their terms using familiar technology with simple processes and an ability to engage with and share content they're interested in.

Lack of Transparency: We live in a time where people can track their lunch being delivered in real-time, but generally have no ability to see the impact of their giving. People expect transparency over how their donations are used, specifically around charity accountability; administrative costs; and the direct impact for the receiver.

Poor Donor Experience: Most current donation platforms do not allow donors to see the impact of their donation. This lack of emotional connection leads to disengagement and donor attrition.

Exploitation of Generosity: Charities have no way of validating their beneficiaries authenticity or their access to support from other charities, leading to inefficient use of funds.

1.4. Objectives

The main objective of this work is to develop a blockchain technology that can help to solve the fundraising industry issues to rebuild trust and increase the desire to give by providing donors with: Decreased fundraising costs so more money makes it to the cause; Transparency, accountability and control over how their money is used; Evidence of the direct impact their donation had.

Therefore, the specific objectives of this project are:

- Creation of a blockchain smart-contract solution to secure trust and accountability functions to an immutable public ledger that allows tracing the usage of funds;
- Creation of the Proof of Impact - that holds the payments until validation is provided by the receiver using their Unique Digital Identities (UDID);
- Creation of an ERC-20 standard token that can be stored and transferred to or from any ERC-20 compatible wallet for better traceability of transactions and costs.

CHAPTER 2: BIBLIOGRAPHIC REVIEW

2.1. Field Related Work

There are many articles and papers that seek to use blockchain to solve old industry problems [16], as well as the impact on the use of technologies to assist the financial traceability and cost reduction of the current banking system [17], yet it's a relatively new concept, being coined for the first time in 2008 as a peer-to-peer electronic cash system [18], although techniques of how to time-stamp a digital document appeared way before in 1991 [19], as a propellant of we know today as a smart contract.

Crypto-philanthropy [20] has been taken up by some high-profile charitable organizations in recent years. In 2017, for example, the global philanthropic organization Fidelity Charitable received the equivalent of $69 million in cryptocurrency donations [20]. The Blockchain Charity Foundation (BCF) is another notable example of crypto-philanthropy. The BCF is a not-for-profit organization aiming to transform philanthropy through the use of a decentralized charity platform [28].

2.2. Concepts and Relevant Techniques

2.2.1. Cryptocurrency Technology

In year 2008, an individual or group writing under the name of Satoshi Nakamoto published a paper entitled "Bitcoin: A Peer-To-Peer Electronic Cash System" [18]. This paper described a peer-to-peer version of the electronic cash that would allow online payments to be sent directly from one party to another without going through a financial institution. Bitcoin was the first realization of this concept [19]. Now word cryptocurrencies is the label that is used to describe all networks and mediums of exchange that uses cryptography to secure transactions-as against those systems where the transactions are channeled through a centralized trusted entity.

Bitcoin is the most popular example that is intrinsically tied to blockchain technology [17]. It is also the most controversial one since it helps to enable a multibilliondollar global market of anonymous transactions without any governmental control. Hence it has to deal with a number of regulatory issues involving national governments and financial institutions [17].

2.2.2. Blockchain Technology

A blockchain is essentially a distributed database of records or public ledger of all transactions or digital events that have been executed and shared among participating parties [20]. Each transaction in the public ledger is verified by consensus of a majority of the participants in the system. And, once entered, information can never be erased. The blockchain contains a certain and verifiable record of every single transaction ever made. To use a basic analogy, it is easy to steal a cookie from a cookie jar, kept in a secluded place than stealing the cookie from a cookie jar kept in a market place, being observed by thousands of people.

It has the potential to revolutionize the digital world by enabling a distributed consensus where each and every online transaction, past and present, involving digital assets can be verified at any time in the future [19]. It does this without compromising the privacy of the digital assets and parties involved. The distributed consensus and anonymity are two important characteristics of blockchain technology.

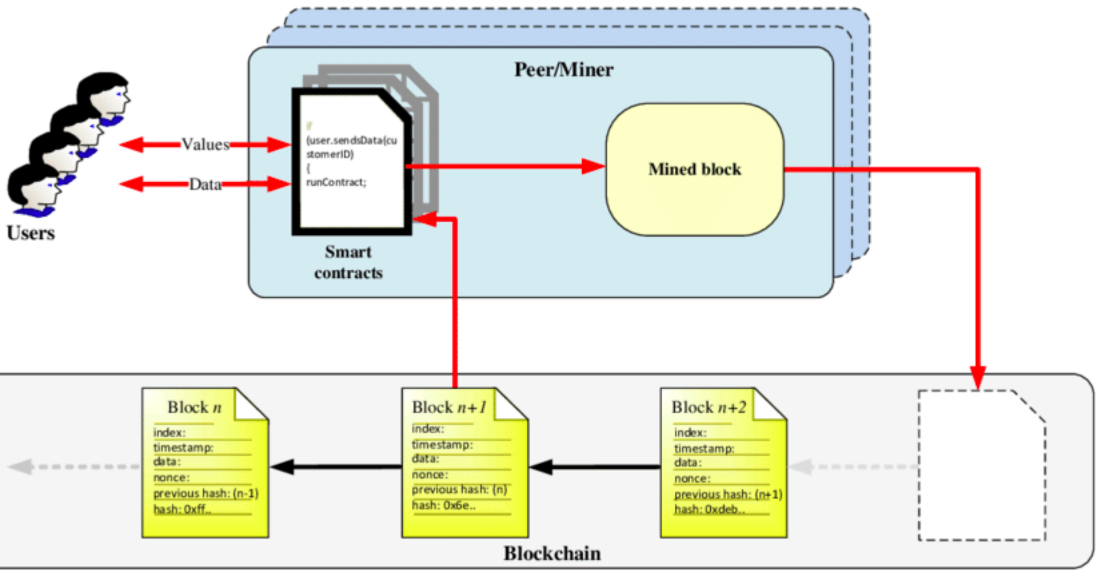


Figure 1 - How blockchain and smart contract work [22].

2.2.3. Ethereum and Smart Contracts

Many researchs show several ways to use Bitcoin's scripting language to support interesting applications [21], such as an escrowed payment transaction. We've also seen how Bitcoin script is somewhat limited, with a small instruction set that isn't Turing−complete [20].

What if, instead of needing to launch a new system to support every application, we built a cryptocurrency that could support any application we might dream up in the future? This what Turing−completeness is all about: to the best of our knowledge, a Turing−complete programming language lets you specify any functionality that is possible to be specified by any other computer [20].

Ethereum is an ambitious alternative coin that aims to provide a Turing−complete programming language for writing scripts or "contracts" [22]. The term smart contract was first used to describe the use of computer systems (or other automated means) to enforce contracts. As an example, you could think of a vending machine as a mechanical smart contract that enforces an agreement between you and the machine's owner involving the purchase of a candy bar [21].

In Ethereum, a contract is a program that lives on the blockchain. Anybody can create an Ethereum contract, for a small fee, by uploading its program code in a special transaction. This contract is written in bytecode and executed by a special Ethereum−specific virtual machine, usually just called EVM [22]. Once uploaded, the contract will live on the blockchain. It has its own balance of funds, other users can make procedure calls through whatever API the program exposes, and the contract can send and receive money.

2.3. Why Blockchain?

2.3.1. Potential Benefits

The main reason that motivated the choice of the blockchain to carry out the work was the scalability that this can promote to the work of the NGOs. NGOs are limited in terms of growth because their work is often “ant" scale, small, and they need to check each and everry step of the process in person. The blockchain takes away the human check factor and promotes more speed in the processes of the organization. Besides, there are some notable advantages for charitable organizations and donors [17], which includes:

Global and decentralized: most blockchain networks present high levels of decentralization, meaning that they do not need to rely on a centralized government or other institution. Thus, funds can move directly from donors to charities, and the decentralized nature of blockchain makes it uniquely suitable for international transactions, with the potential to simplify the way charities are managed, automating parts of the process and reducing the overall costs by requiring fewer intermediaries.

2.3.2. Limitations

Despite the potential advantages, there are some potential concerns to be considered when adopting this type of philanthropy [17]:

Volatility: besides stablecoins, most cryptocurrencies are being traded on highly volatile markets, and often suffer large swings in value.

Security: if the private keys that give access to donated funds are lost, there is no way to recover them. Likewise, if the keys are not managed and secured properly, a malicious entity may end up accessing the wallets and stealing the funds.

Public awareness and understanding: most people find blockchain quite difficult to explain, and many potential donors don’t understand the basics of cryptocurrencies well enough to trust the system or make use of it for charitable donations.

CHAPTER 3: DEVELOPMENT

3.1. Initial Considerations

In this chapter will be presented the chosen methods to solve the problems of trust and accountability within donations in fundraising and crowdfunding for social causes through extended blockchain technology, as well as it's applicability in the Brazilian nonprofit sector, through a voluntary and partner NGO, STH.org.br [Appendix B], that set their system available in order to implement and test our solution.

3.2. Description of Overall System Function

Donators give online using their local money and the system converts the donation to cryptocurrency, transaction fees are low-cost and transparent, the app will show you the exchange rate so you know exactly what you'll send, process known as swap. The system escrow locks the value of these fund using smart contracts, waiting for Proof of Impact validation from the receiver. Once validated, the smart contract is unlocked and the system sends the payment for the supplier and rewards the giver.

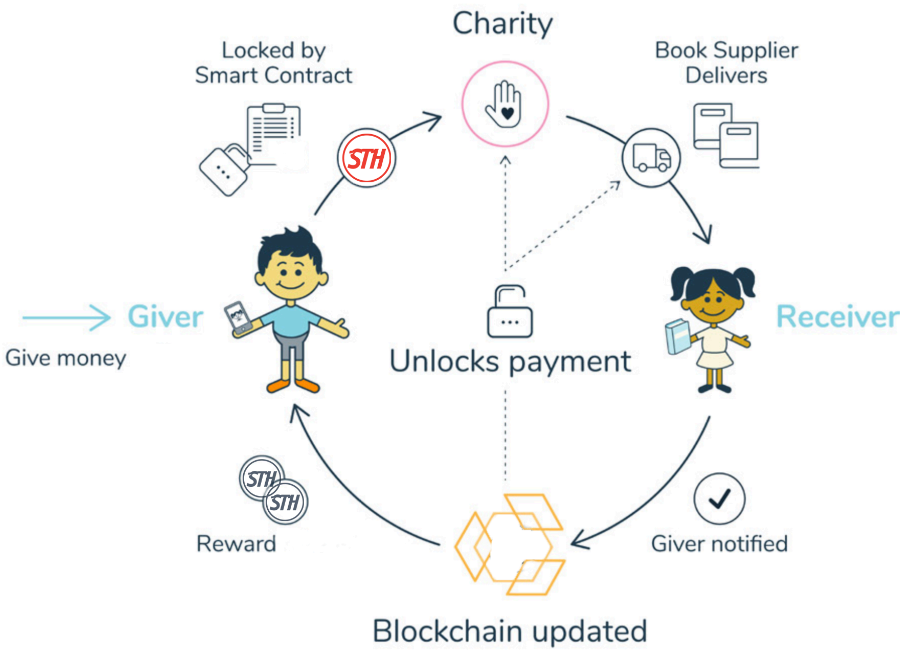


Figure 2 - How the system works.

The system works through a program built with Solidity [Annex A:1], an object-oriented, high-level programming language for implementing smart contracts, that manages and generates smart contracts fort each new donation. The payments are in the meanwhile holded into an ERC-20 wallet as a token format, to improve the traceability of the money itself and future integration.

A biometric authentication is required when the funds reached the goal or needed to be withdraw to pay the supplier and/or receiver, let's say, a hospital surgery and so on. This step was currently developed using Swift iOS Touch ID [Annex A:2] and it's consider a very simple format of what we model as an UDID.

Finally, the system is compiled and deployed by using Truffle [Annex A:3], which is a testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), and then runned with Ganache [Annex A:4] that simulates an Ethereum localhost blockchain.

3.2.1. Unique Digital Identities (UDID)

UDID's are not a new ideia [23]. Although, for the context of this work, we had to remoduled existent open source applications [Annex A:5] to create an electronic identity solution using biometric integrations to ensure only legitimate receivers can register and remain on the system, so that UDID's can assist with:

- Fraud prevention (no duplicate receivers);
- Better resource allocation for charities;
- Identifying and personifying undocumented people.

Any donations to that receiver will be associated with their UDID and recorded on the public ledger.

3.2.2. Proof of Impact

Proof of Impact is the name we gave to validating the impact of the action, which was one of the major concerns as shown in the introduction. This system provides givers with positive reinforcement, transparency and accountability over the usage of their funds. Do not misunderstand this concept with the blockchain inherited Proof of Work [24].

Proof of Impact - child medical treatment example:

1. Ana decides to contribute in the purchase of a surgery for a child in Brazil through the our system in her local currency which is held in a smart contract.

2. The local hospital (supplier) is notified of the surgery and details. They know the funds are guaranteed and pending until the procedure is done.

3. Multiple biosignatures confirm the deliver of the medical treatment to the receiver, instantly unlocking the smart contract funds to payout the supplier .

4. The system virtually rewards Ana for her action and send her the greetings that the surgery has been done.

3.3. Description of Activities Performed

3.3.1. Smart Contract System

All the donation management and routines were created using the programming language Solidity [Annex A:1] and OpenZeppelin [Annex A:5], which is an open source library for secure smart contract development with pre implementations modules of ERC-20 stardards token. Solidity was influenced by C++, Python and JavaScript and is designed to target the Ethereum Virtual Machine (EVM) [25].

The first step was to design the Donation Wallet code [Source-Code 1], as you can see below, a contract in the sense of Solidity is a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain. You can think of it as a single slot in a database that can be queried and altered by calling functions of the code that manages the database. In the case of Ethereum, this is always the owning contract. And in this case, the functions ‘donate’ and ‘ balance’ can be used to modify or retrieve the value of the variable.

Source-Code 1: Script of the Donation Wallet using Solidity

```
pragma solidity ^0.4.4;

contract DonationWallet is Ownable {
  using SafeMath for uint256;

  function donate(uint _amount, string _projectName) public onlyOwner {
    address projectAddress = projectCatalog.getProjectAddress(_projectName);
    require(projectAddress != address(0));
    ERC20 token = Project(projectAddress).getToken();

    token.approve(projectAddress, _amount);
    Project(projectAddress).donateFromWallet(_amount);
  }

  function refund(ERC20 _token, uint _amount) public onlyOwner {
    _token.transfer(owner, _amount);
  }

  function balance(ERC20 _token) public view returns(uint256){
    return _token.balanceOf(this);
  }

}
```

After establishing the wallet, we proceed to develop the code that created the core logics in the act of donating [Appendix A: Source-Code 3], developing the functionalities around a ‘New Donation’ and to payout a ‘Payee’, which represents our Receiver or Supplier of the cause.

To construct the model of the proof of impact, we built a Claim Verification process [Appendix A: Source-Code 4] based on how investors in the crowdsale verify their identity and claims [26], and then integrating with the biometric authetication.

And finally integrating with an ERC-20 standard token within our smart contract system [Appendix A: Source-Code 5].

3.3.2. Biometric Authentication

To develop our UDID model, we've built a mobile application with Swift on Apple iOS system, by eliminating the necessity of a proprietary mechanism, using the fingerprints library [Annex A: 6] if the iOS device supports Touch Id recognition. Due to the availability of our equipaments, we've choosen iOS for now, keeping in mind that the process is very similar to build for Android later.

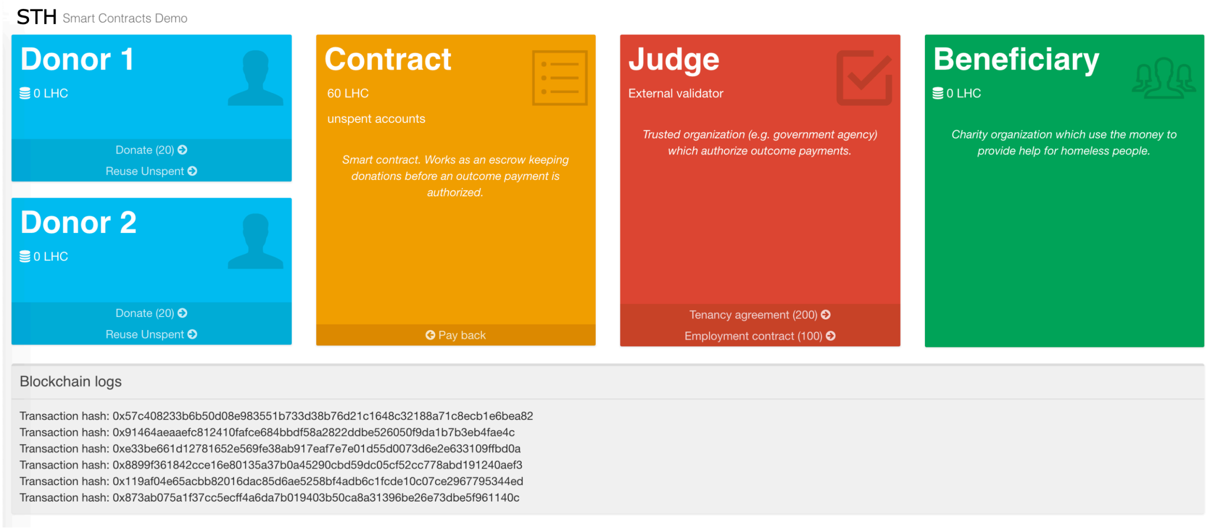
Source-Code 2: iOS Biometric Local Authentication of Touch ID using Swift

```
//  Biometrics Authentication

import UIKit
import LocalAuthentication

class ViewController: UIViewController {

    @IBAction func touchAuthenticateUser(_ sender: Any) {

        // Device can use biometric authentication
        if context.canEvaluatePolicy(
            LAPolicy.deviceOwnerAuthenticationWithBiometrics,
            error: &error) {
            context.evaluatePolicy(
                .deviceOwnerAuthenticationWithBiometrics,
                localizedReason: self.strAlertMessage,
                reply: { [unowned self] (success, error) -> Void in
                    DispatchQueue.main.async {
                        if( success ) {
                            //Fingerprint recognized
                            self.goToNextVC()

                        } else {
                            //If not recognized then
                                self.notifyUser("Error",
                                                err: strMessage)
                        }
                    }
            })
        }
    }

}
```

The code developed above works with Apple Touch ID (iPhone X, Xs, XR, XsMax) equipaments and other Touch ID having devices. Blocking the automatic authentication with device passcode on multiple failed attempts and predefining error handling when recognition fails.

3.3.3. Application Environment

In order to test our application, we've also built a front-end to simulate the functionalities of the system and it's responses inside the blockchain and the smart contract. By using Truffle and Ganache combined, we created a local blockchain for Ethereum in development mode so that we could deploy our contracts and start to run tests. It is available as both a desktop application and a command-line tool (formerly known as the TestRPC) [Annex A:7].



Smart contract. Works as an escrow keeping donations before an outcome payment is authorized.

Trusted organization (e.g. government agency) which authorize outcome payments.

Charity organization which use the money to provide help for homeless people.

Blockchain logs

```
Transaction hash: 0x57c408233b6b50d08e983551b733d38b76d21c1648c32188a71c8ecb1e6bea82
Transaction hash: 0x91464aeaaefc812410fafce684bbdf58a2822ddbe526050f9da1b7b3eb4fae4c
Transaction hash: 0xe33be661d12781652e569fe38ab917eaf7e7e01d55d0073d6e2e633109ffbd0a
Transaction hash: 0x8899f361842cce16e80135a37b0a45290cbd59dc05cf52cc778abd191240aef3
Transaction hash: 0x119af04e65acbb82016dac85d6ae5258bf4adb6c1fcde10c07ce2967795344ed
Transaction hash: 0x873ab075a1f37cc5ecff4a6da7b019403b50ca8a31396be26e73dbe5f961140c
```

Figure 3 - First application launched.

Each time a donor's money is used to pay for a goal, that donor receives a notification, and can track, at any time, what goals their gift has helped to achieve, and how much is left on their donation balance.

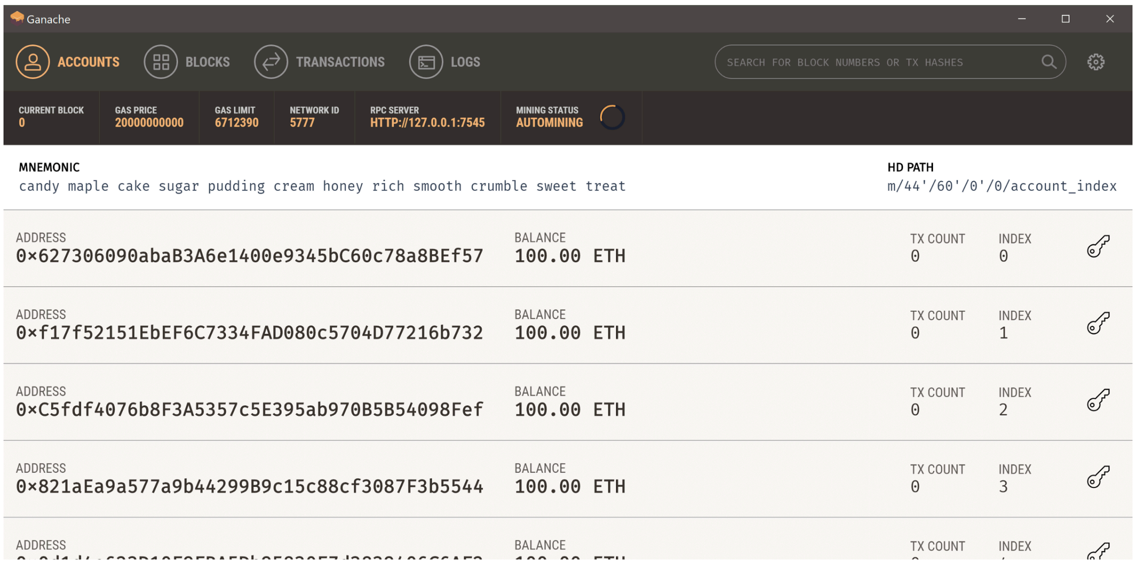
After each donation is processed, a transaction hash is recorded at the blockchain logs. Each interaction with the system generates a transaction at the ledger, recorded by it's public key [27] of who and/or which account has lead to this change.

3.4. Results Obtained

The initial goal was to develop a system, focused on it's traceability power and low costs of operation - since small amount donations are accepted, that could be able to make a donation for a direct cause, allows tracing the usage of those funds, proving the impact.

To validate the minimum viable product (MVP), we have counted with the help of several institutions and organisations [Appendix B], and tested the whole process in a real scenario, involving a donator, the listed NGO, a hospital, a child and their legal guardians.

It was also possible to see it fully functioning, although running locally and not deployed to Ethereum live environment, we could see some inconsistencies with ERC-20 token, and even the optionality of removing it from the first version of the system or proceeding to the creation of our own token that could be listed on exchanges to operate.



```
CURRENT BLOCK  GAS PRICE    GAS LIMIT  NETWORK ID  RPC SERVER              MINING STATUS
0              20000000000  6712390    5777        HTTP://127.0.0.1:7545   AUTOMINING

MNEMONIC                                                                    HD PATH
candy maple cake sugar pudding cream honey rich smooth crumble sweet treat  m/44'/60'/0'/0/account_index

ADDRESS                                       BALANCE     TX COUNT  INDEX
0×627306090abaB3A6e1400e9345bC60c78a8BEf57    100.00 ETH  0         0

ADDRESS                                       BALANCE     TX COUNT  INDEX
0×f17f52151EbEF6C7334FAD080c5704D77216b732    100.00 ETH  0         1

ADDRESS                                       BALANCE     TX COUNT  INDEX
0×C5fdf4076b8F3A5357c5E395ab970B5B54098Fef    100.00 ETH  0         2

ADDRESS                                       BALANCE     TX COUNT  INDEX
0×821aEa9a577a9b44299B9c15c88cf3087F3b5544    100.00 ETH  0         3

ADDRESS                                       BALANCE     TX COUNT  INDEX
```

Figure 4 - System functioning in a localhost Ethereum blockchain with Ganache.

In comparison to other studies in the field, this work uses techniques and algorithms relatively new, with little evidence similar to the proposed and achieved joint, but with deficiencies already explained by previous studies [17] although it fulfills its initial purpose.

3.5. Metrics Evaluation

Our main objective of the work was to develop a technology capable of increasing the desire to donate, following some metrics and criteria, as shown in the Table 2 below. We met and compared our solution with traditional solutions to see if we could be more efficient as we promised.

Services	Fundraising Costs	Money Control	Impact Evidence
Doare	8% [31]	N/A	N/A
Gofundme	7.9% [31]	N/A	Random Check
Paypal for nonprofits	2.9% + \$0.30 [31]	N/A	N/A
Our Solution	1.39%	Smart-Contract	Proof of Impact

Table 2 - Comparison between traditional fundraising and our solution.

As we use the crypto-currency and currency-crypto exchange form known as swap, we managed to reduce transaction costs ranging from 2.9% to 8% in the market, to 1.39% [30], which represents a drop of at least 50% of traditional costs, plus transparency and impact verification.

3.6. Difficulties, Limitations and Future Work

We must highlight here the very real challenges of creating such an arbitration system. Human disputes always entail at least some degree of subjectivity, making it difficult to resolve them in an unbiased way on-chain, without recourse to a centralised system such as a government court system.

Several Ethereum-based projects are developing decentralised arbitration solutions [28], and we will seek to develop partnerships as we build and iterate the platform.

If the ideia is to continue the developing of the system with a go-to-market strategy, an Initial Coin Offering (ICO) [29] would be recommended, and it validates the use of our own token, allowing greater functionality and integration of the system with fiat and cryptocurrency exchanges, and the donor could give in whichever currency they want.

A second iteration would be the creation of the Proof of Need, which is an extension of the Proof of Impact, it would allow cross-collaboration from charities without the need to share databases. By developing the project we noticed some recipients could exploit the system by registering with multiple charities – at the expense of those who have genuine needs. So if charities had a system which validated that their recipient was not already receiving similar assistance elsewhere, their money would go further in addressing the charity's cause and helping those who actually need it.

CHAPTER 4: CONCLUSION

4.1. Considerations about the Project Developed

On a technical level, much of the innovation created by this project stands in the full tokenisation of social impact funding, delivery and reporting, and the synergies that this tokenisation allows for. We expect that, in many cases, other organisations or individuals may build (and monetise) equal and/or complementary services. We welcome these developments, and will provide all possible assistance to ensure they can make a good use of the donation system and thrive. The objectives of the project was achieved, showing that it is possible for donations to be transparent and fun.

4.2. Risks and Uncertainties

4.2.1. Regulatory Risk

Blockchain technology allows new forms of interaction and it is possible that certain jurisdictions will apply existing regulations on, or introduce new regulations addressing, blockchain technology based applications, which may be contrary to the current setup of any smart contract and which may, result in substantial modifications to the system. Additionally, regulation of the proposed activities of the system, including without limitation the system itself, is currently uncertain. It is not known what regulatory framework the system and associated activities will be subject to.

4.2.2. Unanticipated Risks

The underlying software application and software platform (i.e. the Ethereum blockchain) may be exposed to attacks by hackers or other individuals including, but not limited to, malware attacks, denial of service attacks, consensus based attacks, Sybil attacks, smurfing and spoofing. Any such successful attacks could result in theft or loss of contributions, adversely impacting the ability to develop the system and derive any usage or functionality from it. Furthermore, because the system is based on open-source software, there is a risk that a third party may intentionally or unintentionally introduce weaknesses or defects into the core infrastructure, which could negatively affect the application.

REFERENCES

[1] PHILANTHROPY AUSTRALIA. Fast facts and statistics on giving in Australia. 2016. Accessed: 2018-06-09. Available at: <http://www.philanthropy.org.au/tools-resources/fast-facts-and-stats/>

[2] IDIS. Pesquisa Doação Brasil. 2016. Accessed: 2019-03-09. Available at: <https://idis.org.br/pesquisadoacaobrasil/wp-content/uploads/2016/10/PBD_IDIS_Sumario_2016.pdf>

[3] CHINA DEVELOPMENT BRIEF. Annual Report on China's Charitable Donations in 2016 released in Beijing. 2017. Accessed: 2018-09-12. Available at: <http://www.chinadevelopmentbrief.cn/news/annual-report-on-chinas-charitable-donations-in-2016-released-in-beijing/>

[4] CHARITIES AID FOUNDATION. CAF UK Giving 2017. 2017. Accessed: 2018-09-23. Available at: <https://www.cafonline.org/docs/default-source/about-us-publications/caf-uk-giving-web.pdf?sfvrsn=8>

[5] NATIONAL PHILANTHROPIC TRUST. Charitable Giving Statistics. 2017. Accessed: 2018-09-25. Available at: <https://www.nptrust.org/philanthropic-resources/charitable-giving-statistics/>

[6] NCCS. Quick Facts About Nonprofits. 2016. Accessed: 2018-09-28. Available at: <http://nccs.urban.org/data-statistics/quick-facts-about-nonprofits>

[7] DAILY MAIL. Charities are spending the majority of donations chasing more money with as little as 17 percent of funds going to those in need. 2017. Accessed: 2018-10-02. Available at: <http://www.dailymail.co.uk/news/article-4589268/Queensland-charities-spending-revealed-investigation.html>

[8] ACCC. Research into the Commission-based Charity Fundraising Industry in Australia. 2017. Accessed: 2018-10-09. Available at: <https://www.accc.gov.au/system/files/Research%20into%20the%20Commission-based%20Charity%20Fundraising%20Industry%20in%20Australia.pdf>

[9] PHILANTHROPY AUSTRALIA. Giving Australia 2016. 2016. Accessed: 2019-01-03.

Available at: <http://www.philanthropy.org.au/images/site/blog/Giving_Australia_2016_Individuals_Fact_Sheet.pdf>

[10] BLACKBAUD. The Next Generation of American Giving. 2013. Accessed: 2019-01-05. Available at: <https://www.blackbaud.com/nonprofit-resources/generational-giving-report-infographic>

[11] NY TIMES. Connections to a Cause: The Millennial Way of Charity. 2016. Accessed: 2019-01-06. Available at: <https://www.nytimes.com/2016/11/06/giving/connections-to-a-cause-the-millennial-way-of-charity.html>

[12] THIRD SECTOR. Half of people give up when trying to donate by mobile. 2013. Accessed: 2019-01-08. Available at: <https://www.thirdsector.co.uk/half-people-give-when-trying-donate-mobile/fundraising/article/1188581>

[13] SOFII. How changing fundraising's culture will make donors feel great about their giving and the difference they make. 2017. Accessed: 2019-01-10. Available at: <http://sofii.org/images/Articles/The-Commission-on-the-Donor-Experience/FULL-PROJECT-SUMMARY.pdf>

[14] NFPSYNERGY. Public levels of trust in charities and Royal Family increase, but political parties still bottom of pile, new research shows. 2012. Accessed: 2019-01-12. Available at: <https://nfpsynergy.net/public-levels-trust-charities-and-royal-family-increase-political-parties-still-bottom-pile-new-1#downloads>

[15] ONNÉE, S.; RENAULT, S. Crowdfunding: principles, trends and issues . F. Xavier Olleros and Majlinda Zhegu. Handbook of Research on Digital Transformations, Editions Edward Elgar, pp.313-334, 2015, 9781784717759.

[16] CROSBY, M.; NACHIAPPAN, C.; PATTANAYAK, P.; VERMA, S.; KALYANARAMAN, V. BlockChain Technology: Beyond Bitcoin. Sutardja Center for Entrepreneurship & Technology Technical Report. 2015. Accessed: 2019-01-12. Available at: <https://scet.berkeley.edu/wp-content/uploads/BlockchainPaper.pdf>

[17] VIGNA, P.; CASEY, M. J. The Truth Machine: The Blockchain and the Future of Everything. St. Martin's Press, 2018. ISSN 9781250114600.

[18] NAKAMOTO, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Accessed: 2019-01-23. Available at: <https://bitcoin.org/bitcoin.pdf>

[19] HABER, S.; STORNETTA, W. S. How to Time-Stamp a Digital Document. Journal of Cryptology. 1991. ISSN 1432-1378.

[20] NARAYANAN, A.; BONNEAU, J.; FELTEN, E.; MILLER, A.; GOLDFEDER, S. Bitcoin and Cryptocurrency Technologies. 2016. Accessed: 2019-01-26. Available at: <https://d28rh4a8wq0iu5.cloudfront.net/bitcointech/readings/princeton_bitcoin_book.pdf>

[21] Jovovic, Ivan & Husnjak, Sinisa & Forenbacher, Ivan & Maček, Sven. 5G, Blockchain and IPFS: A General Survey with Possible Innovative Applications in Industry 4.0. 2018. 10.4108/eai.6-11-2018.2279695.

[22] Alharby, Maher & van Moorsel, Aad. (2017). Blockchain Based Smart Contracts : A Systematic Mapping Study. 125-140. 10.5121/csit.2017.71011.

[23] Bakre, Akshay & Patil, Nikita & Gupta, Sakshum. (2017). Implementing Decentralized Digital Identity using Blockchain. 10.54121/csit.2016.72045.

[24] Gervais, Arthur & Karame, Ghassan & Wüst, Karl & Glykantzis, Vasileios & Ritzdorf, Hubert & Capkun, Srdjan. (2016). On the Security and Performance of Proof of Work Blockchains. 3-16. 10.1145/2976749.2978341.

[25] DANNEN, C. (2017). Solidity Programming. 10.1007/978-1-4842-2535-6_4.

[26] PROPERTY2CHAIN. ERC 725 and ERC 735—identity and claims. 2017. Accessed: 2019-01-26. Available at: <https://www.property2chain.io/Whitepaper.pdf>

[27] Zheng, Zibin & Xie, Shaoan & Dai, Hong-Ning & Chen, Xiangping & Wang, Huaimin. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. 10.1109/BigDataCongress.2017.85.

[28] ANTONOPOULOS, A. M.; WOOD, G. Mastering Ethereum: Building Smart Contracts and DApps. O'Reilly, 2019.

[29] ALEXANDER, M. Initial Coin Offerings - Ico's: Introduction and Guide to Ico Token Sales. 2017. ISBN: 9781975660949.

[30] KANG, M. W. Efficiency of Foreign Exchange and Its Related Derivatives Market: Evidence From Korea. International Journal of Financial Research. 2019. 10.5430/ ijfr.v10n2p92.

[31] MOTYLSKA-KUZMA, A. The Cost Of Crowdfunding Capital. 2016. 10.1145/3212734.3212736.

APPENDIX A - SOURCE CODES

Source-Code 3: Program that manages the donation

```
pragma solidity ^0.4.4;

// STH Donations.sol

contract Donation {

  // Instantiate a variable to hold the account address of the contract administrator
  address public owner;

  // Create a data structure to reperesent each of the participants.
  struct Payee {
    // If the payee can administer their account.
    bool status;
    // The amount relative to aggregate weight that a payee will receive.
    uint weight;
    // A record of the amount held for the payee.
    uint balance;
  }

  // Create an associative arrays with account address as key and payee data sturcture as value.
  mapping(address => Payee) public payees;
  // Create an array like mapping to behave as an index of addreesses
  mapping (int8 => address) public payeesIndex;
  // Keep note of total number of participants in the system so we can iterate over index.
  int8 public payeesIndexSize;

  // Declare events for actions we may want to watch
  event NewDonation(address indexed donator, uint amt);
  event Transfer(address indexed from, address indexed to, uint amt);
  event PayeeAction(address indexed payee, bytes32 action);
  event Withdrawal(address indexed payee, uint amt);
  event OwnerChanged(address indexed owner, address indexed newOwner);
  event ContractDestroyed(address indexed contractAddress);

  // Constructor
  function Donation() {
    // Set the address of the contract deployer to be owner.
    owner = msg.sender;
    payees[owner].status = true;
    payees[owner].weight = 10;
    payeesIndex[0] = owner;
    payeesIndexSize = 1;
  }

  // Check if current account calling methods is the owner.
  modifier isOwner() {
    if (msg.sender != owner) throw;
    _;
  }

  // Check if current account calling methods is a valid payee of the contract.
  modifier isPayee() {
    if (payees[msg.sender].status != true) throw;
    _;
  }

  // Aggregate all payee weights.
  function getTotalWeight() private returns (uint) {

    int8 i;
    uint totalWeight = 0;

    for (i=0;i<payeesIndexSize;i++) {
       if (payees[payeesIndex[i]].status == true) {
         totalWeight += payees[payeesIndex[i]].weight;
       }
    }

    return totalWeight;
  }

  // Function which will accept donations.
  function deposit() payable {

  if (msg.value == 0) throw;
    int8 i;
    uint totalWeight = 0;

    totalWeight = getTotalWeight();
    // Update account balances for all payees.
    for (i=0;i<payeesIndexSize;i++) {
       if (payees[payeesIndex[i]].status == true) {
         uint divisor = (totalWeight / payees[payeesIndex[i]].weight);
         payees[payeesIndex[i]].balance = msg.value / divisor;
       }
    }

    NewDonation(msg.sender, msg.value);
  }

  // Add a new payee to the contract.
  function addPayee(address _payee, uint _weight) isOwner returns (bool) {

    payees[_payee].weight = _weight;
    payees[_payee].status = true;
    payeesIndex[payeesIndexSize] = _payee;
    payeesIndexSize++;

    PayeeAction(_payee, 'added');
  }

  // Amend payee weight.
  function updatePayeeWeight(address _payee, uint _weight) isOwner {
    payees[_payee].weight = _weight;
  }

  // Disallow an account address from acting on contract.
  function disablePayee(address _payee) isOwner returns (bool) {
    if (_payee == owner) throw; // Dont lock out the main account
    payees[_payee].status = false;
    PayeeAction(_payee, 'disabled');
  }

  // Allow an account address from acting on contract.
  function enablePayee(address _address) isOwner {
    payees[_address].status = true;
    PayeeAction(_address, 'enabled');
  }

  // Transfer some Ether available to withdraw to another account.
  function transferBalance(address _from, address _to, uint amount) isOwner
{
    if (payees[_from].balance < amount) throw;
    payees[_from].balance -= amount;
    payees[_to].balance += amount;
    Transfer(_from, _to, amount);
  }


  function getBalance(address _address) isPayee returns (uint) {
    return payees[_address].balance;
  }


  function getWeight(address _address) isPayee returns(uint) {
    return payees[_address].weight;
  }


  function getStatus(address _address) returns(bool) {
    return payees[_address].status;
  }


  // Change ownership of the contract.
  function transferOwner(address newOwner) isOwner returns (bool) {
    if (!payees[newOwner].status == true) throw;
    OwnerChanged(owner, newOwner);
    owner = newOwner;
  }

  // Destroy the contract and pay out all enabled members.
  // Any outstanding value will be transferred to owner.
  function kill() payable isOwner {
    int8 i;
    address payee;

    for (i=0;i<payeesIndexSize;i++) {
      payee = payeesIndex[i];
      if (payees[payee].balance > 0 ) {
        if (payee.send(payees[payee].balance)) {
          Withdrawal(payee, payees[payee].balance);
        }
      }
    }

    ContractDestroyed(this);
    selfdestruct(owner);
  }
}
```

Source-Code 4: Program that verifies the identity

```
pragma solidity 0.4.4;

import './ClaimHolder.sol';

contract ClaimVerifier {

  event ClaimValid(ClaimHolder _identity, uint256 claimType);
  event ClaimInvalid(ClaimHolder _identity, uint256 claimType);

  ClaimHolder public trustedClaimHolder;

  function ClaimVerifier(address _trustedClaimHolder) public {
    trustedClaimHolder = ClaimHolder(_trustedClaimHolder);
  }

  function checkClaim(ClaimHolder _identity, uint256 claimType)
    public
    returns (bool claimValid)
  {
    if (claimIsValid(_identity, claimType)) {
      emit ClaimValid(_identity, claimType);
      return true;
    } else {
      emit ClaimInvalid(_identity, claimType);
      return false;
    }
  }

  function claimIsValid(ClaimHolder _identity, uint256 claimType)
    public
    constant
    returns (bool claimValid)
  {
    uint256 foundClaimType;
    uint256 scheme;
    address issuer;
    bytes memory sig;
    bytes memory data;

    // Construct claimId (identifier + claim type)
    bytes32 claimId = keccak256(trustedClaimHolder, claimType);

    // Fetch claim from user
    ( foundClaimType, scheme, issuer, sig, data, ) =
_identity.getClaim(claimId);

    bytes32 dataHash = keccak256(_identity, claimType, data);
    bytes32 prefixedHash = keccak256("\x19Ethereum Signed Message:\n32",
dataHash);

    // Recover address of data signer
    address recovered = getRecoveredAddress(sig, prefixedHash);

    // Take hash of recovered address
    bytes32 hashedAddr = keccak256(recovered);

    // Does the trusted identifier have they key which signed the user's
claim?
```

```
    return trustedClaimHolder.keyHasPurpose(hashedAddr, 3);
  }

  function getRecoveredAddress(bytes sig, bytes32 dataHash)
      public
      view
      returns (address addr)
  {
      bytes32 ra;
      bytes32 sa;
      uint8 va;

      // Check the signature length
      if (sig.length != 65) {
        return (0);
      }

      // Divide the signature in r, s and v variables
      assembly {
        ra := mload(add(sig, 32))
        sa := mload(add(sig, 64))
        va := byte(0, mload(add(sig, 96)))
      }

      if (va < 27) {
        va += 27;
      }

      address recoveredAddress = ecrecover(dataHash, va, ra, sa);

      return (recoveredAddress);
  }

}

```

Source-Code 5: Program structure of ERC-20 token verification

```
pragma solidity 0.4.4;

contract ERC20 {

    event ClaimRequested(uint256 indexed claimRequestId, uint256 indexed claimType, uint256 scheme, address indexed issuer, bytes signature, bytes data, string uri);    event ClaimAdded(bytes32 indexed claimId, uint256 indexed claimType, address indexed issuer, uint256 signatureType, bytes32 signature, bytes claim, string uri);
    event ClaimAdded(bytes32 indexed claimId, uint256 indexed claimType, uint256 scheme, address indexed issuer, bytes signature, bytes data, string uri);
    event ClaimRemoved(bytes32 indexed claimId, uint256 indexed claimType, uint256 scheme, address indexed issuer, bytes signature, bytes data, string uri);
    event ClaimChanged(bytes32 indexed claimId, uint256 indexed claimType, uint256 scheme, address indexed issuer, bytes signature, bytes data, string uri);

    struct Claim {
        uint256 claimType;
        uint256 scheme;
        address issuer; // msg.sender
        bytes signature; // this.address + claimType + data
        bytes data;
        string uri;
    }

    function getClaim(bytes32 _claimId) public constant returns(uint256 claimType, uint256 scheme, address issuer, bytes signature, bytes data, string uri);
    function getClaimIdsByType(uint256 _claimType) public constant returns(bytes32[] claimIds);
    function addClaim(uint256 _claimType, uint256 _scheme, address issuer, bytes _signature, bytes _data, string _uri) public returns (bytes32 claimRequestId);
    function removeClaim(bytes32 _claimId) public returns (bool success);
}
```

APPENDIX B - INSTITUTIONAL PARTNERS

Below are the institutional partners that have made this project feasible:

STH SomosTodosHeróis

INSTITUTO C.
CRIANÇA • CUIDADO • CIDADÃO

HOSPITAL
MUNICIPAL • INFANTIL
MENINO JESUS

BLOCKCHAIN
AT BERKELEY

Pantas and Ting
Sutardja Center
for Entrepreneurship & Technology
Berkeley Engineering

Griffith
UNIVERSITY

THE UNIVERSITY OF CALIFORNIA BERKELEY
LET THERE BE LIGHT
1868

ANNEX A - DOCUMENTATION

[1] <https://solidity.readthedocs.io/en/v0.5.7/> : Solidity;

[2] <https://swift.org/documentation/> : Swift;

[3] <https://truffleframework.com/docs/truffle/overview> : Truffle;

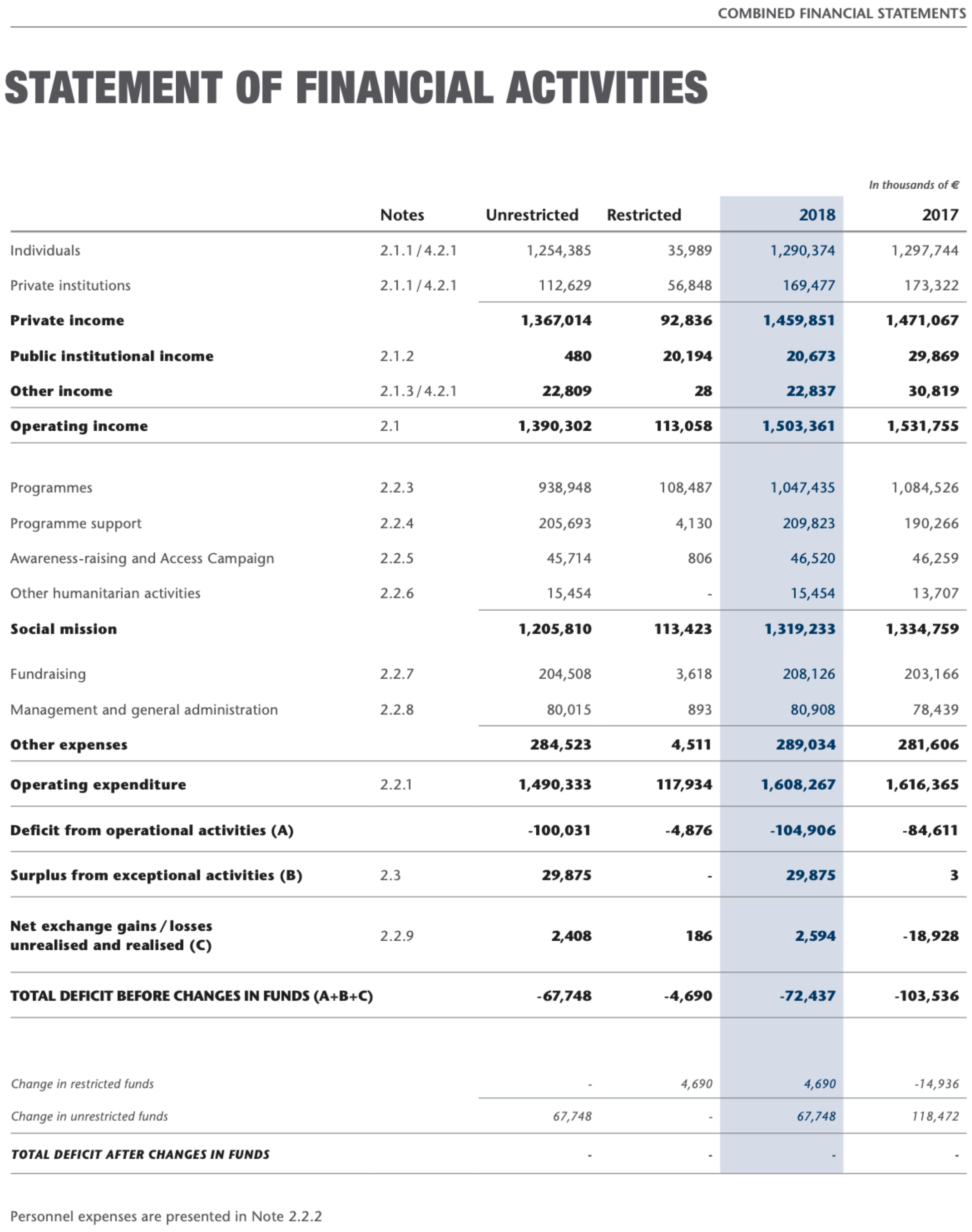
[4] <https://truffleframework.com/docs/ganache/overview> : Ganache;

[5] <https://github.com/OpenZeppelin/openzeppelin-solidity> : OpenZeppelin;

[6] <https://github.com/rushisangani/BiometricAuthentication> : BiometricAuthentication;

[7] <https://docs.nethereum.com/en/latest/ethereum-and-clients/test-rpc/> : TestRPC;

ANNEX B - MSF FINANCIAL REPORT 2018



COMBINED FINANCIAL STATEMENTS

STATEMENT OF FINANCIAL ACTIVITIES

In thousands of €

	Notes	Unrestricted	Restricted	2018	2017
Individuals	2.1.1 / 4.2.1	1,254,385	35,989	1,290,374	1,297,744
Private institutions	2.1.1 / 4.2.1	112,629	56,848	169,477	173,322
Private income		**1,367,014**	**92,836**	**1,459,851**	**1,471,067**
Public institutional income	2.1.2	**480**	**20,194**	**20,673**	**29,869**
Other income	2.1.3 / 4.2.1	**22,809**	**28**	**22,837**	**30,819**
Operating income	2.1	**1,390,302**	**113,058**	**1,503,361**	**1,531,755**
Programmes	2.2.3	938,948	108,487	1,047,435	1,084,526
Programme support	2.2.4	205,693	4,130	209,823	190,266
Awareness-raising and Access Campaign	2.2.5	45,714	806	46,520	46,259
Other humanitarian activities	2.2.6	15,454	-	15,454	13,707
Social mission		**1,205,810**	**113,423**	**1,319,233**	**1,334,759**
Fundraising	2.2.7	204,508	3,618	208,126	203,166
Management and general administration	2.2.8	80,015	893	80,908	78,439
Other expenses		**284,523**	**4,511**	**289,034**	**281,606**
Operating expenditure	2.2.1	**1,490,333**	**117,934**	**1,608,267**	**1,616,365**
Deficit from operational activities (A)		**-100,031**	**-4,876**	**-104,906**	**-84,611**
Surplus from exceptional activities (B)	2.3	**29,875**	**-**	**29,875**	**3**
Net exchange gains / losses unrealised and realised (C)	2.2.9	**2,408**	**186**	**2,594**	**-18,928**
TOTAL DEFICIT BEFORE CHANGES IN FUNDS (A+B+C)		**-67,748**	**-4,690**	**-72,437**	**-103,536**
Change in restricted funds		*-*	*4,690*	*4,690*	*-14,936*
Change in unrestricted funds		*67,748*	*-*	*67,748*	*118,472*
TOTAL DEFICIT AFTER CHANGES IN FUNDS		*-*	*-*	*-*	*-*

Personnel expenses are presented in Note 2.2.2

Source: Medecins Sans Frontieres. International Financial Report. 2018. Accessed: 2019-06-02. Available at: <https://www.msf.org/sites/msf.org/files/2019-05/msf-financial-report-2018.pdf>