UNIVERSITÀ DI TRENTO

Department of Information Engineering and Computer Science

Ethix Project Report

ethix

Team members	
Brentan Simone	239959
Costalonga Matteo	239960
Reichert Alex	239961
Farina Pietro	249889

Academic Year 2023/2024

Contents

Introduction

Charity campaigns serve as organised efforts aimed at raising funds or resources for specific causes. These campaigns, often initiated by individuals, corporations, or small businesses, play a crucial role in mobilising resources for charitable purposes. However, traditional charity models frequently face challenges related to **transparency, trust, and efficiency**. Donors often struggle with a lack of visibility into how their contributions are used, leading to diminished trust in charitable organisations and, consequently, reduced participation.

The *Ethix* project addresses these challenges by leveraging **blockchain technology** to enhance the transparency, security and scalability of charity campaigns. By utilising a decentralised platform, *Ethix* ensures that donations are not only securely managed, but also transparently tracked from the moment they are made until they reach the intended beneficiaries. A key feature of this system is **user-controlled donations**, where contributors have direct control over the allocation of campaign funds through the scanning of QR codes associated with purchased products.

Security within the platform is achieved through the usage of **Commit-Reveal Randomness** (CRR) and **asymmetric key cryptography** for secure codes generation (*tokens*), ensuring that all transactions are both secure and verifiable. Additionally, the system is designed with **scalability** in mind, particularly in the generation and management of tokens, which enables the actual transfer of donations. This approach allows to efficiently handle a large number of transactions, ensuring that the system remains responsive and reliable as the number of users and campaigns grows.

Through these innovations, *Ethix* not only addresses the inefficiencies of traditional charity models, but also builds a stronger connection between donors and the causes they support. By providing a **transparent, secure and scalable platform** for charitable giving, the system aims to restore trust in the charity sector and encourage greater participation in charitable activities.

This report provides an in-depth overview of the design, development and implementation of the *Ethix* platform, with a focus on its innovative approach to **charity campaigns and user-controlled donations**. It highlights the technological advancements that drive the platform's functionality, allowing donors to have direct control over how their contributions are used within specific campaigns. The report also explores the implications of using blockchain in the charity sector, considering both the potential benefits and challenges that may arise. By providing a detailed analysis of the system's architecture and operational processes, this paper aims to demonstrate how blockchain can effectively address the limitations of traditional charitable models and create a more equitable and efficient system for managing donations. Through this approach, the project seeks to restore trust in charitable organisations, encouraging more people to contribute to the causes they care about and ultimately enhancing the positive impact of charitable activities worldwide.

1 Problem discussion

This chapter will explore the key challenges faced by the charity sector, first presenting an overview of traditional charity problems and then moving on to the more specific definition of *Charity Campaigns* and analysis of their problems, including issues such as lack of transparency, risk of mismanagement, trust concerns and inefficiencies in fund distribution.

1.1 Problem overview

The **traditional charity sector** faces a multitude of challenges that can affect the trust and efficiency of charitable activities. These issues not only influence negatively the beneficiaries, but also discourage potential donors from contributing, ultimately hindering the overall impact of charitable efforts.

The traditional charity model faces numerous obstacles that hinder its efficiency, transparency and ability to build lasting trust with users. These challenges not only affect how funds are managed, but also impact the relationship between donors and beneficiaries, often leaving both parties dissatisfied.

Typically, the major issues traditional charity suffers from:

- **Lack of transparency**: The current charity system suffers from a lack of transparency when it comes to the management and allocation of funds, leaving no insights and no feedback in how their contributions are used. This lack of visibility not only diminishes trust, but also discourages potential donors from engaging with charitable causes, ultimately reducing the overall impact of charitable efforts.

- **Lack of security and efficiency**: The involvement of multiple intermediaries and administrative layers introduces inefficiencies and delays, which can slow down the distribution of funds. Moreover, these complex processes increase the risk of funds being mismanaged or misappropriated, further reducing the confidence in the system.

- **High operational costs**: Many charitable organisations face high administrative and operational costs, which reduce the actual amount of money that reaches beneficiaries. Donors may be disillusioned to learn that a significant portion of their contributions is used to cover overhead expenses, rather than going directly to the cause they intended to support.

- **Lack of trust**: The cumulative effect of these issues – lack of transparency, security concerns and high costs – leads to a broader lack of trust in the charity system. Without clear and consistent feedback or assurances that their contributions are making a tangible difference, donors are less likely to continue supporting charitable causes.

1.2 Charity campaigns

A *Charity Campaign* is an organised event designed to raise funds or resources for a specific cause. Various models of charity campaigns exist, but this report focuses on a model where contributions are made through the purchase of products.

In fact, in this paper, campaigns are typically initiated by donors, which may include corporate entities or small businesses, such as bakeries, that allocate the initial donations. These campaigns are promoted through social media or advertising, thus increasing greater reach and better visibility. Individuals, who wish to support the cause, participate by purchasing eligible products sold by the donor, thereby indirectly contributing to the success of the campaign.

However, as seen for traditional charities, this type of campaign suffers from several problems ranging from lack of transparency to miscommunication errors, as can be seen in section 1.3. In particular, a detailed study was conducted in order to better understand these issues.

1.3 Case study: Ferragni-Balocco and the Pandoro-Gate

The so-called "Pandoro-Gate", involving the influencer Chiara Ferragni and the Balocco company, is a media and judicial case born from their collaboration for the promotion of a special pandoro during Christmas 2022. The controversy was sparked by the news that part of the proceeds from the sales of Pandoro, which Ferragni claimed it **would be donated** to the Regina Margherita Hospital in Turin for research on osteosarcoma and Ewing's sarcoma, had in fact already been donated by Balocco in May 2022, **before** the **promotional campaign**. This revelation led to a sanction by the Antitrust, which **fined** Ferragni and Balocco for **unfair commercial practices**.

Case Description

The special edition of Balocco's traditional Christmas cake featured Ferragni's logo and cost 9 euros, more than double the normal price. The packaging claimed *"Chiara Ferragni and Balocco support the Regina Margherita Hospital in Turin, financing the purchase of a new machine [...]"*. The information on the label led buyers to believe that there was a direct correlation between the purchase of the pandoro and the donation to the hospital, thus explaining the 154% increase in price. The cakes sold were approximately 290 thousand bringing a gross proceeds of 2 million and 700 thousand euros, while the donation already made in May was equal to 50 thousand euros. In December 2023, the news that the donation had already been made and the proceeds from the pandoro were not intended for the hospital, sparked public disapproval and an investigation by the Milan Public Prosecutor's Office into both Ferragni and Balocco for aggravated fraud. Furthermore, following the appeal presented by the consumer associations, the AGCM fined Balocco for 420 thousand euros and the Ferragni companies for a total of 1 million and 75 thousand euros, justifying the sanction for unfair commercial practice by misleading marketing.

Outcomes

This scandal raises important questions about transparency and ethics in marketing and demonstrates a great sensitivity and attention on the part of consumers, especially in the field of charity campaigns. A correct representation of charitable initiatives is a fundamental requirement, both for the involvement of donors and because depending on the State it may be enforced by law. A lack of transparency may result in a significant backlash: both economic, considering the fines and the companies that have interrupted the contracts with the influencer; and advertising as in the case of over one hundred thousand followers lost on the various social platforms.

2 Solution

In this chapter, a comprehensive solution to the *Charity Campaigns* challenges identified in the previous part is presented. First, the general idea of the proposed solution is presented. The focus is then shifted on how the blockchain can be utilised to address potential issues within the charity sector. Lastly, the proposed solution, referred as *Ethix*, is outlined in its core concepts and value proposition, highlighting its main benefits over other solutions.

2.1 Linking Purchases to Donations

One of the most significant issues in traditional charity campaigns, as highlighted in the "Pandoro-Gate" case, is the lack of a transparent and direct link between consumer purchases and the resulting donations. Consumers are often led to believe that their purchase directly benefits a charitable cause, yet the details of how and when these donations occur are frequently opaque, leading to **mistrust and dissatisfaction** when discrepancies arise. This disconnection undermines the credibility of both the charity and the businesses involved, as seen with the backlash faced by Ferragni and Balocco.

The proposed solution focuses on addressing this critical problem by establishing a clear, verifiable connection between the act of purchasing a charity-linked product and the actual donation process. Through the *Ethix* platform, a system is introduced where the donation is not just promised, but is actively initiated and completed by the end user at the point of purchase. This is achieved through the use of **QR-coded tokens** embedded in the products, that once scanned they complete the donation to the designated beneficiary. This system ensures that the donation only occurs if the product is purchased and scanned, thereby directly linking the consumer's action to the charitable outcome.

By placing the power of the donation in the hands of the consumer, *Ethix* not only enhance transparency, but also rebuild trust in the charity process, allowing consumers to have a **tangible, verifiable role** in the donation process. This approach eliminates the ambiguity often associated with charity-linked purchases, promoting a stronger, more transparent relationship between donors, consumers and beneficiaries.

2.2 Leveraging blockchain for charity

Blockchain technology offers a potential solution to various challenges within the charity sector. Its **transparency**, **immutability** and **public accessibility** can be leveraged to mitigate well known issues such as lack of trust, inefficiency and fraudulent activities in charitable operations.

When implementing blockchain technology in the charity sector, it is essential to clarify the distinction between private and public blockchain networks. Public blockchains, such as Ethereum, are often preferred for charitable applications due to their **transparency**, **security** and **broader user base**. They ensure that all transactions are visible and verifiable by **anyone**, which significantly enhances accountability and promotes **trust** among donors and beneficiaries. The decentralised nature of these networks further strengthens security by minimising the risk of single points of failure. Moreover, the **immutability** of smart contracts guarantees that once the conditions are set, they remain fixed, thereby maintaining the integrity of the charitable process. Despite these advantages, it is important to acknowledge that public blockchains may face **limitations in scalability** compared to private networks, as the larger number of participants can lead to slower transaction processing times and higher computational demands.

This emphasis on immutability and transparency makes blockchain technology particularly advantageous over traditional databases. Specifically, this system ensures that all transactions are securely recorded and cannot be altered. This means that donations and fund allocations remains **unaltered** and **accessible**, effectively addressing integrity and accountability concerns.

2.3 Ethix

The *Ethix* solution represents an innovative approach for enhancing the **security** and **transparency** of charitable operations by leveraging the **public blockchain** technology. At its core, *Ethix* utilises blockchain's features to create a more reliable platform for charitable activities.

2.3.1 Concepts

Generally, the charity campaigns started within the *Ethix* platform work by allowing the end users to actually perform the donations instead of the donor. Although the *donor* is the one actually allocating the donation, the final users are the one in charge of it by buying an eligible product, sold by the donor, and scanning a QR code. The strong point of *Ethix* is, in fact, its feature of making the final users be in control of the donation transfer.
The most important aspect of the system is the concept of **Token** and how it is used inside the platform. When scanning a QR Code, a process of *Token redeeming* takes place, by retrieving the unique value saved inside of it and performing the donation.

To use the service, both donors and beneficiaries must register on the platform, with beneficiaries also requiring to be first verified by a platform administrator.
Donors wishing to make a contribution can initiate a campaign by defining the number of tokens and the verified beneficiary and allocating the entire initial deposit. The tokens, represented as QR codes, are distributed to people by **embedding** them in the donors' products. This method not only **incentivises** product purchases, but also directly **links** the donation process to the consumer's buying decision and places **full responsibility** on end-users.
At the end of a campaign, the number of redeemed tokens is counted and the initial deposit is accordingly divided, with the final resulting donation being distributed to the beneficiary and the remaining portion being refunded to the donor.

2.3.2 Value proposition canvas

Value proposition *Ethix* provides a more transparent and decentralised approach to manage charity campaigns, making the user in control of the actual donation. Traditional systems often lack transparency, causing donor distrust and visibility issues for beneficiaries. Addressing these problems, the platform offers real-time monitoring, clear campaign progress status and ensures donations are used appropriately through features like:

- **QR codes**: enabling the actual donation execution. When scanning a QR Code, its unique token is redeemed and the donation is performed.

- **Blockchain security**: allowing for public transparency and immutability of transactions, enhancing the security of donations.

- **Beneficiary verification**: is performed in order to make sure only trusted beneficiaries can be the target of charity campaigns.

Reference: *Value Proposition Canvas*

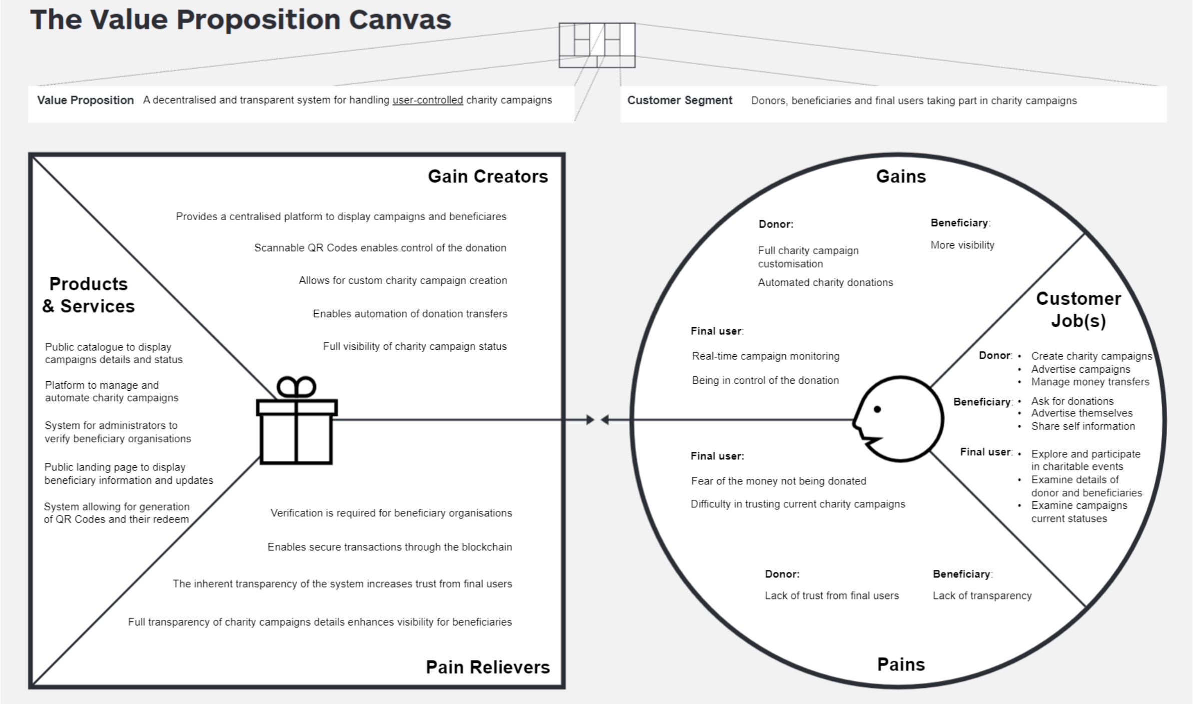
The Value Proposition Canvas
Value Proposition A decentralised and transparent system for handling user-controlled charity campaigns
Customer Segment Donors, beneficiaries and final users taking part in charity campaigns
Gain Creators
Provides a centralised platform to display campaigns and beneficiares
Scannable QR Codes enables control of the donation
Allows for custom charity campaign creation
Enables automation of donation transfers
Full visibility of charity campaign status
Products & Services
Public catalogue to display campaigns details and status
Platform to manage and automate charity campaigns
System for administrators to verify beneficiary organisations
Public landing page to display beneficiary information and updates
System allowing for generation of QR Codes and their redeem
Verification is required for beneficiary organisations
Enables secure transactions through the blockchain
The inherent transparency of the system increases trust from final users
Full transparency of charity campaigns details enhances visibility for beneficiaries
Pain Relievers
Gains
Donor:
Full charity campaign customisation
Automated charity donations
Beneficiary:
More visibility
Final user:
Real-time campaign monitoring
Being in control of the donation
Customer Job(s)
Donor: • Create charity campaigns • Advertise campaigns • Manage money transfers
Beneficiary: • Ask for donations • Advertise themselves • Share self information
Final user: • Explore and participate in charitable events • Examine details of donor and beneficiaries • Examine campaigns current statuses
Final user:
Fear of the money not being donated
Difficulty in trusting current charity campaigns
Donor:
Lack of trust from final users
Beneficiary:
Lack of transparency
Pains

Figure 2.1: Value proposition canvas

Customers segment The primary actors targeted by this system are donors, beneficiaries and final users involved in charity campaigns.

- **Donors:** Individuals or organisations that want to create, advertise and manage charity campaigns efficiently. They require a platform that ensures their contributions are used appropriately and additionally helps in the donation process, by automating the transfer of money.
- **Beneficiaries:** Individuals or organisations in need of donations. They look for increased visibility and a trustworthy platform to ensure that donations are received transparently.
- **Final users:** People interested in exploring, participating in or monitoring charitable activities. They demand trust, transparency, ease of access to detailed information about ongoing campaigns and being in control over the donation process.

Market The market trends indicate a consistent increase in total charitable giving over recent years. As can be seen in figure 2.2, Americans gave a record of almost $485 billion to charities in 2021, with donations likely to continue increasing. This potentially suggests a corresponding growth in the number of charity campaigns, highlighting the relevance of the proposed solution to current market needs. In particular, the following points highlight respectively why this platform is necessary and how it operates to address current market needs:

- The platform addresses the lack of transparency, control and trust in current charity systems. It uses decentralised technologies to improve transparency and user experience, addressing donor concerns and enhancing beneficiary visibility.
- The platform allows users to examine details about ongoing campaigns and beneficiaries, automating donations and ensuring transparency with QR codes and blockchain security.

It offers campaign management, real-time monitoring and simplifies beneficiary verification, addressing the need for greater transparency and efficiency in charity donations.

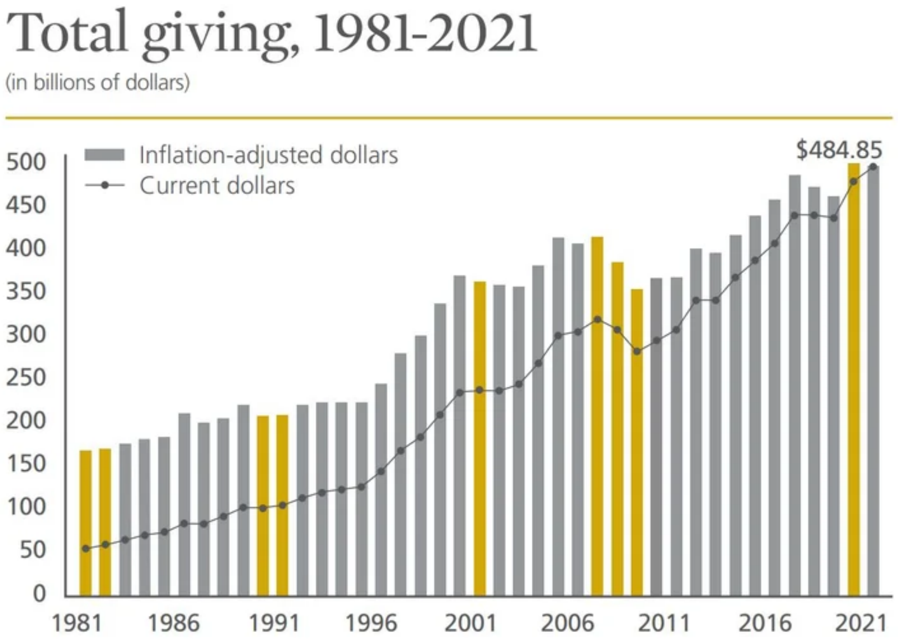

Figure 2.2: Total giving for charities from 1981 to 2021

2.3.3 Goals and focuses

Ethix is designed with several key goals and focuses in mind to optimise its effectiveness and impact in the charity sector.

Flexibility A primary goal is to offer a platform that can adapt to a variety of charitable campaigns. This flexibility enables the system to support a **wide range** of fundraising activities and meet the needs of diverse beneficiaries, making it well-suited for many charitable organisations.

Scalability Scalability is a critical consideration, as the platform must efficiently manage a **high volume** of token redeeming transactions. To address the associated costs and performance challenges, the design of the proposed solution has been optimised to guarantee scalability.

Benefits The solution is designed to offer advantages for both donors and beneficiaries. For donors, it provides a **transparent** and **secure** mechanism for contributing, while also **involving** end users in the donation process. Beneficiaries, instead, gain access to a **reliable** and **accountable** system for receiving funds.

Security The platform employs blockchain technology to create a secure environment where transactions are safeguarded against tampering and fraud. Additionally, a range of cryptographic techniques is employed to ensure the system remains tamper-proof. Smart contracts are provided with comprehensive security measures, including **Commit Reveal Randomness** for token generation and **asymmetric key cryptography** for validating **token signatures**.

3 Architecture

In this chapter, the architecture of *Ethix* is explained and analysed, giving general information about key components and participants of the application. Additionally, each component is later explained in details to better understand its role in the process.

3.1 Division of roles

In general, the architecture of the solution has to provide a way for creating charity campaigns, with a list of **unique tokens**, which can be *redeemed* to perform a donation.
In order to make *Ethix* provide these functionalities to its users, the architecture of the solution involves multiple components accomplishing different tasks.
The divided components consists in different applications interacting with each other, and the division is as follows:

- **Blockchain application**: A decentralised application whose role is to store securely and in a trusted way public information regarding the life of a charity campaign.
- **Server application**: A backend application running in one ore more server, connected with a database, which responds to client requests and is tasked with the token generation and signing.
- **Client application**: A frontend application which offers the users of the system a simple way to view and manage charity campaigns, connecting with the server and interacting with a *Wallet* for authenticating the user on the blockchain.

In addition, four types of users have been identified as the main participants interacting with the application:

- *Donor*: The one responsible for the campaign creation and funding.
- *Beneficiary*: The beneficiary of the campaign, which receives money from the donors. They have to be verified by an administrator first.
- *Admin*: An administrator which has the power to verify or block users.
- *User / Buyer*: The users which buy the product and redeem a token to perform a donation.

3.2 Components interaction and campaign life cycle

The *Ethix* architecture of course revolves around the campaigns and their life cycle.
In order to better understand the flow of operations taking place, this section will go through the entire process of a campaign generation, focusing on the interaction of the interested components.

Registration First of all, both the *Donor* and *Beneficiary* needs to register themselves inside the application, by filling out a form. When registering, they need to have a *MetaMask* **wallet** account connected in order to associate their account address.

Verification Then, an *Admin* has to log in and **verify** the beneficiary by checking the entered information and account. This is done in order to make sure the beneficiary is a legitimate organisation. This is needed only the first time the beneficiary registers in the application.

Campaign creation After this verification process, the *Donor* is now able to **create a campaign** by selecting the just verified *Beneficiary* and filling a list of information such as title, starting date, deadline, donation amount, the number of tokens to generate and the target goal to be reached (See the discussion about generated tokens vs target tokens in section 5.4)

Campaign funding Once a campaign is created, the *Donor* needs also to fund it, thus allocating the initial deposit and **generating the list of QR codes** associated with that campaign. Presumably, the production of the eligible products now starts and the QR codes are embedded in the product boxes.

Token redeem When the starting date is reached, the tokens become redeemable. In order to redeem a token, a *User* just needs to buy an eligible product and scan the QR token present in the box (**no wallet is needed**).

Donation claiming When the deadline is reached, the tokens are not redeemable anymore and the *Beneficiary* and *Donor* can respectively claim the **donations** and **refund**. Depending on the amount of redeemed tokens, the total sum allocated for the campaign is either donated or returned to the *Donor* user.

3.3 Blockchain application

The blockchain decentralised application has been made using **Solidity** smart contracts.
Two contracts have been devised: *Charity* and *Campaign*. The first one is responsible for managing and storing the list of campaigns, as well as enabling the verification of *Beneficiaries* to the *Admin*. The *Campaign* contract, instead, is the one responsible for managing a single campaign details, the balance and tokens.

3.3.1 Charity contract

It can be seen as the entry point for the DApp, as all the methods that are required outside are called from this contract. It functions as a **Campaign Factory**, making possible the creation and funding of new campaigns for *Donor* users, as well as managing **authorisation** of users. It also provides methods for the verification of *Beneficiary* users to the **Admin**.

3.3.2 Campaign contract

Corresponds to a single campaign instance, thus containing its details and providing a way to operate on its data. It is responsible for **storing the balance** with the total amount of donations instantiated, as well as managing the **transfers of money**. It also contains the logic for the token validation.

3.4 Server application

According to the devised architecture, the server application plays a pivotal role in the entire process.
Its main duties are:

- Providing **access to information** to the client application, fetching data from both the database and the blockchain. Data consists mainly in user (*Donor* and *Beneficiary*) account information and campaign details. In this way the application leverage the database to avoid excessive requests to the blockchain and make the solution faster and lighter.

- **Authenticating and authorising** incoming client requests, enabling different type of users to perform separate tasks. Through this mechanism, the *Admin* can manage and verify the list of *Beneficiary* users.
- Generating the list of tokens which will be put inside the QR codes. To provide authenticity of the generated tokens, a random wallet is generated and used to sign the list of tokens.
- Redeeming a token after a QR code is scanned. To avoid having the *Buyer* connect a wallet, a **Relayer account**, which is stored securely inside the server, is used on his behalf.

The server application interacts with an **external database** to store basic access information, some campaign details and a list of token hashes (implementation details regarding token generation can be found in 4.4.2).
The credentials of the *Relayer* wallet account are securely saved inside the server application. Even though it provide some risks, it is a necessity in order to enable users to execute **gasless transactions** for token redeeming. In addition, this account is used to **sign transactions** to the blockchain, so that the server data can be authenticated. For more information regarding the *Relayer* account in general, consult the apposite discussion section 5.2.

3.5 Client application

The client application serves as the user interface for interacting with the platform, providing an **intuitive** and **user-friendly** experience for both donors and beneficiaries.

MetaMask integration

A key feature of the client application is its integration with **MetaMask**, a widely-used web3 browser extension. MetaMask serves dual purposes:

- It acts as a **gateway** for the users to interact with the Charity smart contract in the blockchain
- It functions as a **cryptocurrency wallet**, securely storing users' accounts.

For most operations within the platform, connecting to MetaMask is **essential**.

Client as a middleware

The client application also acts as a **middleware** in the system, especially during the campaign **creation** and **funding stages**. In particular, it manages the following tasks:

- During campaign creation, it receives the **signed seed** for token generation directly from the server and performs a **seed-forwarding process** to the blockchain
- During campaign funding, it receives the **signed wallet** for token validation directly from the server and performs a **wallet-forwarding process** to the blockchain

In order to start or fund a campaign, the server application needs to perform some operations and send the resulting parameters to the blockchain. However, in order to make the donor cover for the **transaction fees** required for starting or funding a campaign, they have to pass through the client application.
For this reason, the client can be seen as a middleware, enabling **communication** between **server** and **blockchain** and ensuring the donor is responsible for covering all transaction fees.

4 Implementation

In this chapter, the practical aspects of implementing the proposed solution are examined. The key features of the system are introduced and explained through detailed charts, providing a comprehensive understanding of their functionality and integration.

4.1 General insights

The implementation of the prototype followed three main applications: the client, the server and the blockchain distributed application.

- **Client frontend application** has been developed using the *React* framework for Javascript language.
- **Server backend application** has been developed using the *Node.js* framework also for Javascript language.
- For the **Blockchain DApp** the *Solidity* language has been chosen for writing the smart contracts for the *Ethereum* network.

Additionally, in order to test the functionalities of the system, a local environment has been built by leveraging *Ganache* as a local network provider.

4.1.1 Commit-Reveal Randomness for seed generation

One of the possible ways to generate a random seed on the blockchain, starting from a previously generated seed, leverages the **Commit-Reveal Randomness** (CRR) method.
Essentially, the secure seed is encoded using the block hash of a **future network block number**, which cannot be predicted. Therefore, in order to wait for the current block to be mined, a two-step function is required.
The CRR standard methods are:

- **Hash**: Take the pre-image (seed generated off-chain) and encode it using a hash function.
- **Commit**: Send to the blockchain the commit hash (hash of the off-chain seed) and save it alongside the current block number. Ethix
- **Reveal**: This function can be called only when the block number has changed from before. The reveal method takes the original seed as input, which is hashed and verified with the saved commit hash, and generates the new secure seed starting from it and the current block number.

4.1.2 Random wallet for token authenticity

While moving the generation of tokens off-chain improved scalability of the application and reduced the amount of fees, it added more risks and vulnerabilities to the system. In order to make sure the server is the one that actually generated the tokens for that specific campaign, *Ethix* leverages the asymmetric cryptography, implemented within wallet accounts, to provide a way to authenticate the server. For each campaign, a pair of public and private keys is generated through the web3 library *accounts.create* method.
After generating the list of tokens, the server signs them using the private key of the random wallet account, and then throws away the key. The public key, instead, is signed using the

‘Server account‘ private key and then sent to the contract during the funding process. The public key is saved inside the campaign contract and then discarded everywhere else.
In this way it was possible to move the generation of tokens off-chain while still maintaining a similar level of security.

4.2 User registration and verification

The system supports four distinct types of users: admin, donor, beneficiary and normal user.

Admin The admin role, as of now, is unique, meaning that it is hard-coded in the database. However, this can be easily updated in the future to accommodate additional admins. Additionally, the admin must use a trusted account to operate on the blockchain contract. For practical reason, during the implementation of the prototype, the admin operated using the default *Ganache* deployer account, which also corresponds to the **owner** of the ‘Charity‘ contract. However, it could simply be changed with a mapping of trusted administrators, but this falls outside the scope of the prototype and was ignored. This account grants admins the authority to **verify** or **unverify** beneficiaries on the blockchain, thereby controlling their status within the system.

Donors and beneficiaries They are required to register by filling out a form that collects not only **personal information** and **login credentials**, but also their **bank account**. This information is crucial for the verification process. During registration, they must also connect a **MetaMask account**, which links their identity in the system's database with their blockchain address. After this process, both donors and beneficiaries gain access to a **dedicated dashboard**, allowing them to manage and view the campaigns they are involved in. In particular, donors can **access all the associated tokens** after correctly funding a campaign.

Beneficiaries verification They must undergo a **manual verification process** carried out by the system administrators. This is done to determine the **legitimacy** of the organisation. Upon successful validation, the beneficiary is correctly **recorded into the blockchain**.

Users Regular users don't need to register, as their only role is to **redeem tokens** for the products they purchase which can be performed easily **without any account**.

4.3 Campaign creation

The campaign creation process involves several steps to ensure that all the necessary details are properly **validated** before storing them into the blockchain. In particular, all blockchain interactions in this process are performed client-side, which ensures that the donor is **responsible** for covering the **transaction fees**, rather than the system.

1. **Draft submission and validation:** This **draft process** comprises both the campaign details submission and their verification server-side, which ensures that they are validated before being submitted to the blockchain.

2. **Seed generation and signing:** During the draft process, a **random seed**, referred as *(Sc)*, is also generated and then hashed to create *(HSc)*. This hash will be utilised in the ***Reveal*** phase of the *Commit-Reveal Randomness (CRR)* to generate a **random seed** for the campaign for **tokens generation**. To ensure the authenticity of *(HSc)*, the corresponding **signature** is generated.

3. **Campaign creation on blockchain:** In this step, the submitted data and the signature are properly **validated**, allowing the blockchain to **extract** and **confirm** that the seed was **generated by the server**. Using MetaMask, the donor sends the data to the blockchain, where the campaign is created and stored, with the **seed** and the **block number** securely saved as part of the ***Commit*** phase of the *Commit-Reveal Randomness (CRR)* process.

4. **Campaign creation on database:** Once the campaign is created on the blockchain, it can be recorded in the database. During this process, the database entry is **associated** to the newly created blockchain **campaign address**.

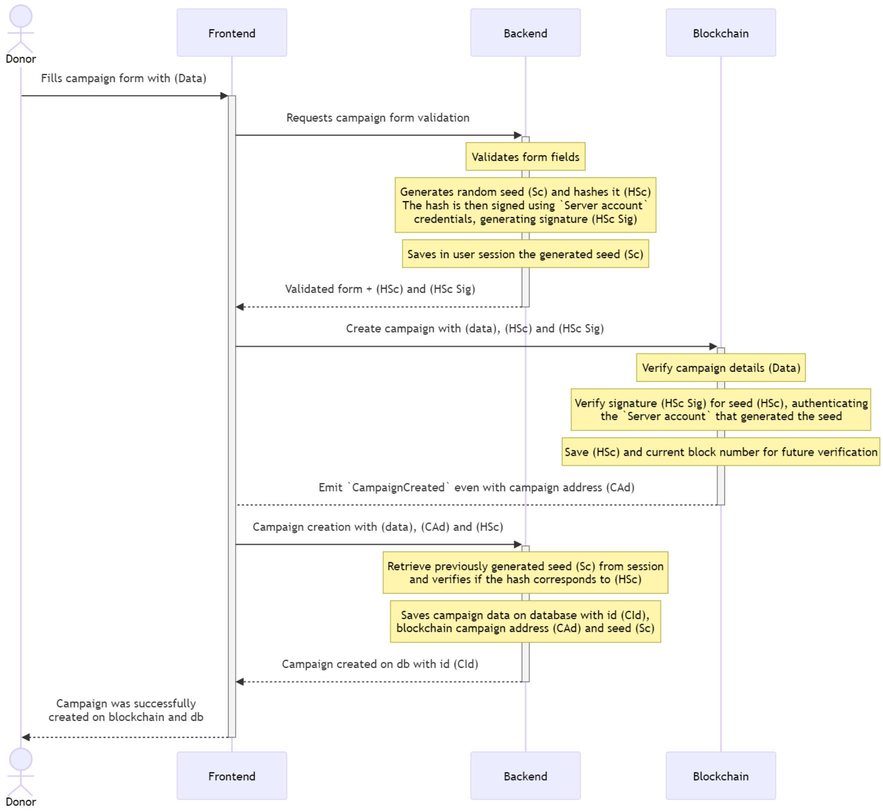
Donor
Frontend
Backend
Blockchain
Fills campaign form with (Data)
Requests campaign form validation
Validates form fields
Generates random seed (Sc) and hashes it (HSc)
The hash is then signed using `Server account` credentials, generating signature (HSc Sig)
Saves in user session the generated seed (Sc)
Validated form + (HSc) and (HSc Sig)
Create campaign with (data), (HSc) and (HSc Sig)
Verify campaign details (Data)
Verify signature (HSc Sig) for seed (HSc), authenticating the `Server account` that generated the seed
Save (HSc) and current block number for future verification
Emit `CampaignCreated` even with campaign address (CAd)
Campaign creation with (data), (CAd) and (HSc)
Retrieve previously generated seed (Sc) from session and verifies if the hash corresponds to (HSc)
Saves campaign data on database with id (CId), blockchain campaign address (CAd) and seed (Sc)
Campaign created on db with id (CId)
Campaign was successfully created on blockchain and db
Donor
Frontend
Backend
Blockchain

Figure 4.1: Campaign creation flow

4.4 Campaign Funding and token generation

In order to simplify and make more secure the system, the campaign funding process and the token generation process are interconnected.
In fact, assigning the tokens generation task to the server application makes the entire platform to be more scalable and reduces transaction fees. For this reason, the process of funding the campaign is the most complex of the entire system.

4.4.1 Campaign funding process

Considering the necessity of generating a random seed blockchain-level, the funding process was separated from the campaign creation function. This was done following the *Commit-Reveal Randomness* approach, where the creation function corresponds to the *Commit CRR* method and the funding function corresponds to the ***Reveal*** *CRR* method. On the client application, this is shown by disabling the button until a new block number, different from the one saved before, is detected from the network.

Therefore, the general steps executed during the funding process are devised as follows:

1. **Random wallet generation** for campaign tokens signing
2. **Confirmation** of the donation and **retrieval** of campaign seed
3. **Authentication and approval** of campaign funding request.

After these steps, the server is ready to generate the list of tokens for the campaign.

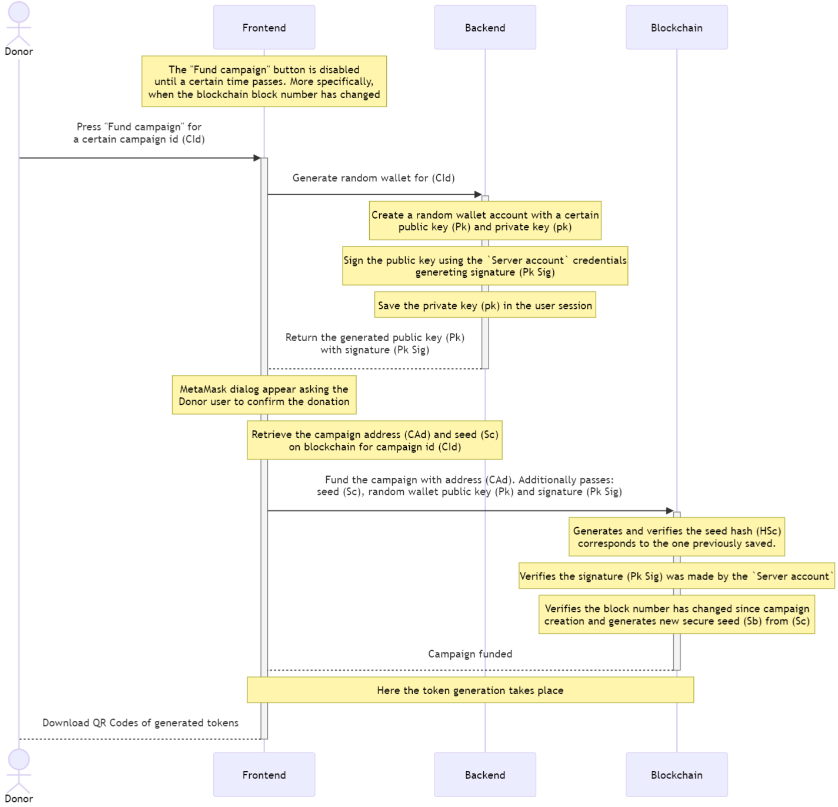
Donor
Frontend
Backend
Blockchain
The "Fund campaign" button is disabled until a certain time passes. More specifically, when the blockchain block number has changed
Press "Fund campaign" for a certain campaign id (CId)
Generate random wallet for (CId)
Create a random wallet account with a certain public key (Pk) and private key (pk)
Sign the public key using the `Server account` credentials genereting signature (Pk Sig)
Save the private key (pk) in the user session
Return the generated public key (Pk) with signature (Pk Sig)
MetaMask dialog appear asking the Donor user to confirm the donation
Retrieve the campaign address (CAd) and seed (Sc) on blockchain for campaign id (CId)
Fund the campaign with address (CAd). Additionally passes: seed (Sc), random wallet public key (Pk) and signature (Pk Sig)
Generates and verifies the seed hash (HSc) corresponds to the one previously saved.
Verifies the signature (Pk Sig) was made by the `Server account`
Verifies the block number has changed since campaign creation and generates new secure seed (Sb) from (Sc)
Campaign funded
Here the token generation takes place
Download QR Codes of generated tokens
Donor
Frontend
Backend
Blockchain

Figure 4.2: Campaign funding flow

Random wallet generation In the first step of the process, a random wallet is instantiated with the its public *(Pk)* and private keys *(pk)*. The public key needs to be sent to the blockchain *Campaign* contract, and for this reason needs to be signed by the 'Server account' with signature *(Pk Sig)* in order to provide the authenticity of the operation. The private key, instead, is temporarily saved in the user session so that it can be used later, after the successful funding operation, to sign the generated list of tokens.

Donation confirmation and data retrieval After generating the random wallet keys, the client application needs to retrieve the previously generated seed *(Sc)* for the campaign, together with the public key *(Pk)* and signature *(Pk Sig)* of the random wallet. Afterwards, the *Donor* is prompted with a *MetaMask* dialog to confirm the donation. After acceptance, the donation amount is taken from its wallet account and transferred to the blockchain., together with the retrieved campaign data.

Authentication and approval of funding On the contract, when the final funding request arrives, it needs to perform a validation and authentication process.
First, the current block is checked and compared to the previously saved one, requiring it to be different. This means the previous block was mined and the ***Reveal*** *CRR* operation can be performed. After confirming the seed hash corresponds to the previously saved commit hash, the random blockchain-level seed *(Sb)* is generated starting from it. Next, the signature of the sent public key is authenticated, requiring the signer to be the *Server account*. After all these verification, the funding operation is approved and the balance is saved in the blockchain *Campaign* contract.

4.4.2 Token generation

After funding the campaign and having shared the random wallet public key to the blockchain contract, the server is ready for the token generation.
The amount of tokens to generate has been set by the *Donor* user during the campaign creation process, as well as the number of target tokens (See the discussion about their difference in section 5.4). In order to make the generated tokens more secure and the system more reliable, the generation of tokens has been divided into a multi-step encoding process, where each time the token is encoded using a different seed. At the end of the encoding process, the final token is signed using the random wallet private key, so that the contract can identify whether the server has generated it or not.

Multi-step encoding process

The multi-step encoding process basically consists of these steps:

1. Generation of tokens (*T1*) starting from an initial seed (*Sc*)
2. Encoding of tokens (*T1.5*) through a list of salts (hashes) saved on the database
3. Encoding of tokens (*T2*) using a secure seed on the blockchain
4. Signing the last round of encoded tokens (*T2*) generating a list of token signatures (*T2 Sig*)

T1 Generation The first step is the simple generation of tokens (*T1*) starting from an initial seed (Sc), which was generated during the campaign creation process by the server and shared with the blockchain contract. Even though this seed is know both server-side, through the database, and blockchain-side, the purpose is different. In fact, the server uses it for generating the initial list of tokens, while in the blockchain it is used by being encoded into a new secure seed (*Sb*) through the **Commit-Reveal Randomness** approach.

T1.5 Encoding In the second phase of the process, the server is first tasked with the generation of a list of token salts (referred to as *T1.5 Salts*). These salts are used to encode the previous tokens *T1* into a new list of tokens (*T1.5*). In order to keep record of which salt was used for each token, the list of salts *T1.5 Salts* is stored in the database together with an hash (*T1 Hash*) of the token it encoded. In this way, during the redeeming process, the salt can be retrieved by calculating the hash of the sent token.

T2 Encoding The next step, instead, involves the encoding of the tokens using the secure seed known by the *Campaign* blockchain smart contract. This seed (*Sb)* was generated directly within the contract using the **CCR** approach. In order to encode the list of tokens (*T1.5*) into a new list of tokens (*T2*), it is necessary to call a smart contract function that encodes it using the seed only known to the contract. While this can operation can be bothersome when the number of tokens to generate increases, it can be performed in batch (See the discussion section for major details 5.3.1).

Signature generation After all these steps, the final list of tokens has been encoded through the multi-step process. The last operation the server does is the signing of these tokens using the secret random wallet private key (*pk*), thus generating a list of token signatures (*T2 Sig*). Then the server discards both the wallet public and private key, while in the blockchain remains saved only the public key for validation. In this way no more token can be generated without generating a new random wallet.

Finally, each token signature (*T2 Sig*) and its initial generative token (*T1*) are encoded in a **JSON Web Token** (*JWT*) to return to the client, which in turns generate the QR Codes to print.

4.5 Token validation

Once a campaign is funded, the associated tokens can be redeemed. This **redeeming process** takes place on a dedicated page of the application, in particular:

- A MetaMask account is **not required**. This design choice was made to **simplify** the user experience, especially for all of those that are not familiar with the blockchain technology.
- The cost of the redeem transaction is handled by a server-side **relayer account** (See the related discussion section 5.2).

Token validation, instead, involves a **reverse process** of the token generation, ensuring that its **authenticity** and **integrity** are validated before the redeeming operation. The general steps of the process are:

1. **Token input:** the token itself is a **JWT** (JSON Web Token) encoded as a QR Code, in which multiple information are recorded such as the token *(T1)*, the signature *(T2 Sig)* and the blockchain campaign id.
2. **Recovering the salt:** The token *(T1)* is extracted from the JWT and its hash is used to **recover** from the database the corresponding **Salt**.
3. **Generating intermediate token:** The *Salt* and the token *(T1)* are then **hashed together** to produce an intermediate value, referred as the token *(T1.5)*.
4. **Blockchain validation:** The token *(T1.5)* along with the signature *(T2 Sig)* are then sent to the blockchain, which performs both **campaign** and **token validation**. In particular, for the token validation the blockchain uses the token *(T1.5)* and the associated campaign seed to generate the token *(T2)*. This is then **compared against** the provided signature to **derive** the signing address, which is finally compared with the wallet specified during the campaign creation to ensure consistency and validity.

Transaction fees drawback The use of blockchain verification enhances **transparency** and **traceability**, making the system more reliable. However, these advantages come together with some other problems also related to the inherent use of the blockchain: the process can be **costly** due to individual transaction fees and may introduce delays, particularly during high network traffic (see more in section 5.3).

Batch Redeeming To address the **high costs** of individual token validation, a batch redeeming process has been employed. This approach allows multiple tokens to be processed in a single transaction, **reducing fees** and improving the overall efficiency (See the apposite discussion section 5.3.1).

Target tokens amount During campaign creation, the *Donor* user specified both the amount of tokens to generate and the amount of target tokens (See difference between the two of them in section 5.4). While the tokens generation process used the first value, the validation mechanism needs the second one. This is because the total donation is reached when the number of QR Codes redeemed is equal to the *target amount*, meaning the full donation has been obtained. When such a case happens, the token redemption is saved only on database and not on blockchain, to save up on gas costs and to prepare for a future implementation of *enhanced target goals*, as explained in the discussion section.

4.6 Campaign conclusion

At the conclusion of a campaign, both donors and beneficiaries can claim their respective parts through their dashboards. This process involves performing a transaction on the blockchain using MetaMask.

To claim a donation or return a refund, the process is very simple:

1. **Count the redeemed tokens:** The system calculates the total number of tokens that have been redeemed during the campaign.
2. **Calculate the donation amount:** The value of the redeemed tokens is then summed up to determine the total donation amount, which is released to the beneficiary.
3. **Return unused tokens:** The unclaimed tokens, representing the unused portion of the initial deposit, are returned to the donor.

This method ensures that all funds are properly allocated and any excess is accurately refunded.

5 Discussions

5.1 Scalable approach for token generation

During the development of Ethix, it was observed that generating a large number of tokens directly on the blockchain posed significant scalability challenges. The process of **creating tokens on-chain** required the generation of unique token IDs and their storage, which severely **limited** the system's ability to handle a high volume of tokens. Specifically, during local testing, the system was able to handle a maximum amount of 200 tokens. Considering that the number of tokens in real campaigns is often much higher, this represents a critical issue.
To address this limitation, the token generation process was **entirely moved** to the server-side. This adjustment significantly **enhanced** the scalability of the system, enabling the creation and management of a huge number of tokens **without overloading** the blockchain. Despite this shift, the **security and integrity** of the tokens were **preserved** through the use of cryptographic techniques. Generating tokens server-side allows the platform to efficiently manage large-scale campaigns involving thousands of tokens, with the redeeming process done off-chain, while still ensuring that each token is securely validated on-chain and recorded at a later stage.

5.2 Server as a Relayer

A **Relayer** is an intermediate actor that facilitates the interactions between users and the blockchain, which is responsible of paying the users transaction fees in their place. This is especially valuable for users who lack the resources to interact directly with the blockchain.
In Ethix, a prototype **server-side Relayer** has been implemented to enable **gas-less transactions**, particularly simplifying the token redeeming process. Although still in the prototype stage, this setup highlights the potential for the integration of external Relayer services.

In real cases, this execution is achieved through **Meta Transactions**, where users can initiate blockchain-related operations without holding the required cryptocurrency for gas fees, as the Relayer covers these costs by submitting signed transactions on their behalf. However, in the last years, **account abstraction** has begun to replace meta transactions, by providing a more flexible and integrated approach for managing transactions and gas fees.

5.3 Gas cost for redeeming process

One of the problems of storing charity campaigns progress and tokens redeem inside the blockchain is the huge number of resulting transactions. Considering the relatively huge cost of a blockchain transaction, the basic solution where everything is saved inside the contract is not possible.
The basic approach, in fact, consists in executing a smart contract transaction to change the blockchain state every time a single token is redeemed. While being the most secure and transparent approach, this mechanism results in excessive transactions fees.

A different solution, involving instead the storage of charity campaigns progress inside an external database, would solve the current issue. However, in such case, the contract will only keep track of basic campaign information, making the usage of the blockchain **questionable**. Although this is a working solution for the problem, it will result in a system relying too much on the security of the server and blockchain, also reducing the overall transparency of the application. Not conforming to the goals of *Ethix*, this solution was discarded.

5.3.1 Redeeming tokens in batch

While totally moving the storage of campaigns progress inside an external database is not accepted, a **hybrid approach** mixing both the basic and database-only solution is an optimal alternative. If calling at each token redeeming the smart contract function consumes a lot of gas, then just reducing the total amount of redeem calls will drastically reduce the costs. For this simple reason, an approach involving the execution of the redeeming function in batches could be implemented.

In such solution, the server will maintain for a certain number of time (e.g. a day) the stored tokens and will only save them to the blockchain when that time passes. Alternatively, the campaign status could be updated when a certain number of redeemed tokens is reached (e.g. 100 tokens), thus making more predictable the cost of the transactions fees. This alternative hybrid solution is the one that was chosen to be implemented inside the prototyped system.

While in the prototype a fixed batch size of 100 tokens has been set, this value should vary case by case, but as this falls outside the scopes of this paper it was not examined in details.

5.3.2 Redeeming tokens gas cost estimation

Although the hybrid approach with the batching process reduces considerably the transaction gas fees for redeeming tokens, it is still not enough to be completely disregarded. Moreover, a method for making the donor pay for these transaction fees should be devised and implemented. At the moment, the prototype has been implemented without taking in consideration the fees paid by the **Server Relayer account** to redeem a token. In fact, when a final user buy an *eligible product* and scan the qr code to redeem the token and confirm the donation, the actual payer of the transaction is the *Server Relayer account* (See related discussion 5.2). However, this costs can not be ignored and should be taken in consideration when funding a campaign, by informing the user of the expected transaction fees.

Despite this fact, the estimation of the overall transaction fees require extensive research on the best way to optimise the token redeeming process, while keeping in consideration both the transaction cost and the maximum delay allowed. The research should include analysis on cost trends throughout the hours and days of the week in order to reduce the gas price and also the computation of the actual fees considering variable token batch size.
In addition, a solution should be found to make this estimated cost be payed in advance by the donor, with some leeway, and at the end of the campaign return the unused funds. Although during the analysis and development of *Ethix* this aspect of the problem was taken in consideration and some research was performed, this topic was eventually left out from the prototype. Considering, in fact, its inherent complexity and the future prospect of implementing an external **Relayer service**, it was evaluated to be out of the scopes of the report.

5.4 Target tokens vs Maximum generated tokens

During the development of the project, a separation has been made between the **amount of target tokens** and the **amount of generated tokens**. The first amount is used to define the number of tokens which needs to be redeemed in order to reach the maximum donation target, meaning that all the allocated initial funds will go to the beneficiary. The number of generated tokens, instead, is greater or equal than the target amount, and is used to define how many tokens and, therefore QR Codes, to generate. This approach allows the total donation to be reached even without the strict requirement of redeeming all the generated tokens.

Considering the system requires the scanning of QR Codes to perform a donation, it can not be expected to be realistically possible the total redemption of all the generated tokens. For this reason, an approach where the target tokens amount is less than the actual generated tokens amount is natural.
Although this approach allows for a simpler management of target goals, it comes with a few challenges. Mainly, when a user redeems a QR Code of a campaign where the target is already reached, in reality its token redemption is not really taken into account as the donation has already been completed. Being a **user-controlled** charity campaign system is a key feature of *Ethix*, and in this way this feature is threatened. However, expecting the redemption of all generated tokens is even more absurd and unreasonable than the previous problem, thus the distinction between *Target tokens* and *Maximum generated tokens* has been born.

In order to restore the lost **user-control**, a possible solution could be to scale up even more the concept of donation target with *enhanced target goals*, adding the possibility to set additional targets when the previous one has been reached. In such a way, the donor needs to allocate new funds when the previous target has been achieved and the token redemptions overflowing from the previous batch would be considered as part of the next one. Although this is a possible valid solution for the problem, this topic still needs to be analysed and revised, as the donor multiple donations requirements may become a bottleneck for the entire platform.

Conclusions

The *Ethix* project successfully demonstrates the potential of blockchain technology to solve and provide a solution for the issues identified for **Charity Campaigns**. Through its implementation, leveraging public blockchain, *Ethix* provides a transparent, secure and efficient platform for managing charitable donations.
The platform's architecture ensures that all transactions are verifiable and that donations are **user-controlled**. This approach not only enhances trust between people and beneficiaries, but also promotes greater transparency and accountability, allowing users to see the direct impact of their contributions.

Moreover, the use of advanced techniques, such as **Commit-Reveal Randomness** and **asymmetric key cryptography**, ensures the security and integrity of the donation process. By involving end-users in the token redeeming process, *Ethix* also promotes greater engagement and accountability, making it a robust solution for modern charitable campaigns.

In conclusion, *Ethix* represents a significant advancement in the field of charitable giving, offering a model that can be scaled and adapted to various types of fundraising activities. As blockchain technology continues to evolve, platforms like *Ethix* are set to play a crucial role in enhancing the transparency, efficiency and impact of charitable initiatives worldwide. The project not only addresses the current problems within the charity sector, but also sets a new standard for how charitable operations can be conducted in a digital and increasingly decentralised world.