DEGREE PROJECT IN TECHNOLOGY,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2021

BlockChain Technologies in the World of Fashion

Nihad Ibrahimli

KTH ROYAL INSTITUTE OF TECHNOLOGY
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Authors

Nihad Ibrahimli nihadibrahimli@gmail.com
Information and Communication Technology
KTH Royal Institute of Technology

Place for Project

Stockholm, Sweden

Examiner

Prof. Mihhail Matskin
Stockholm, Sweden
KTH Royal Institute of Technology

Supervisor

PhD. Shatha Jaradat

Stockholm, Sweden

KTH Royal Institute of Technology

Abstract

Blockchain, a solution for different parties to reach consensus in a peer-to-peer (P2P) networks, allows us to distribute data across different entities. Many areas, including financial, health care, eCommerce, marketing, can benefit from blockchain technology. Influencer marketing also takes benefit from the properties of blockchain technology. Most of the time, social media influencers have an" ideal" body, affecting how the product they are advertising looks on them, which results in increased returns in online shopping. Moreover, existing applications are not involving micro-influencers (10,000 - 100,000 followers) in the campaigns that brands have created. In order to tackle the mentioned problems, a blockchain distributed size and fit application framework is proposed in this thesis. The framework offers a possibility for micro-influencers to participate in the brands' campaigns. However, micro-influencers are not directly involved in communication with brands. Instead, macro-influencers (over 100,000 followers) play a role as a bridge between micro-influencers and brands by creating sub-campaigns of the brands' campaigns. With the involvement of micro-influencers in the proposed framework, they also share their pictures with the same outfit (different body sizes and shapes), which gives potential buyers a better idea of how the same outfit would look on them before they decide to buy. The proposed solution is implemented with smart contracts using Solidity and tested in the Ropsten test network. Testing the smart contract in the Ropsten test network shows that the proposed solution is feasible in terms of financial costs. The proposed framework is not limited to giving the possibility to only brands to advertise products, but it also offers macro-influencers and potential micro-influencers an opportunity to earn a reward in exchange for promoting the products of brands. The whole process starting from applying for a campaign to claiming reward is handled without manual intervention.

Keywords

Blockchain, Smart Contract, Distributed application, Fashion

Sammanfattning

Blockchain, en lösning för olika parter för att nå konsensus i ett peer-to-peer-nätverk (P2P), låter oss distribuera data över olika enheter. Många områden, inklusive ekonomi, hälsovård, e-handel, marknadsföring, kan dra nytta av blockchain-teknik. Influencer-marknadsföring drar också nytta av blockchain-teknikens egenskaper. För det mesta har påverkare på sociala medier en ”idealisk” kropp som påverkar hur produkten de annonserar ser ut på dem, vilket resulterar i ökad avkastning i online-shopping. Dessutom involverar befintliga applikationer inte mikroinfluencer (10 000 - 100 000 följare) i de kampanjer som varumärken har skapat. För att hantera de nämnda problemen föreslås en blockchain-distribuerad applikationsram i denna avhandling. Ramverket erbjuder en möjlighet för mikroinfluencer att delta i varumärkets kampanjer. Mikroinfluencer är dock inte direkt inblandade i kommunikation med varumärken. Istället spelar makroinfluencer (över 100 000 följare) en roll som en bro mellan mikroinfluencer och varumärken genom att skapa underkampanjer av varumärkets kampanjer. Med deltagande av mikroinfluencer i det föreslagna ramverket delar de också sina bilder med samma outfit (olika kroppsstorlekar och former), vilket ger potentiella köpare en bättre uppfattning om hur samma outfit skulle se ut på dem innan de bestämmer sig för att köpa . Den föreslagna lösningen implementeras med smarta kontrakt använder Solidity och testas i Ropsten-testnätverket. Att testa det smarta kontraktet i Ropsten-testnätverket visar att den föreslagna lösningen är genomförbar när det gäller finansiella kostnader. Den föreslagna ramen är inte begränsad till att ge endast varumärken möjlighet att marknadsföra produkter, men den ger också makroinflytande och potentiella mikroinfluencer möjlighet att tjäna en belöning i utbyte för att marknadsföra produkter från varumärken. Hela processen från att ansöka om en kampanj till att få belöning hanteras utan manuellt ingripande.

Nyckelord

Blockchain, Distribuerad applikation, Mode

Acknowledgements

Throughout working on this thesis, I have received a great deal of support and assistance.

I would like to thank my supervisor, PhD. Shatha Jaradat, whose support was invaluable for the whole process of writing this thesis. Your insightful feedback helped me to improve my work to a higher level.

I would like to thank my examiner Professor Mihhail Matskin for taking the time to give me feedback and suggestions to improve the quality of this thesis.

In addition, I also would like to thank my family and my friends who have supported me in dealing with tough phases during these challenging periods.

Contents

Chapter 1

Introduction

Blockchain has become popular lately due to the successful adoption of cryptocurrencies like Ethereum, Bitcoin. As the name states, Blockchain is a sequence of blocks that are chained together. It has a decentralized structure. Blockchain records the history of transactions through the use of decentralization and cryptographic hashing. Blockchain technology provides a solution to transfer digital assets without any intermediaries. Transparency, faster transactions, no intermediary costs in the transactions, and some other benefits of Blockchain make it suitable for companies and businesses. It offers a decentralized, shared, and trusted ledger of transactions that is acceptable by any node in the blockchain network. Smart contracts are the set of business rules defined by the programming code that can be deployed on the blockchain network. This chapter gives an introduction to Blockchain technology and its relation to the project. We describe the problem that this thesis addresses, the context of the problem, the goals of this thesis project, and outlines the structure of the thesis.

1.1 Background

Blockchain technologies have been applied in many different areas, such as financial, health care, insurance, eCommerce, marketing, etc. Since blockchain provides a peer-to-peer network that is not dependent on third-parties, it allows us to create a trustworthy and transparent platform. Smart contracts are lines of programming code that are deployed on the Blockchain. With the help of Smart Contracts on the Blockchain network, we are able to imitate real-world contracts on the digital platform. The main advantage of using smart contracts is to have transparency and faster communication. Once the smart contract is created, it can not be altered. Since smart contracts are digital, programmed versions of the traditional contracts, when they are executed, they process transactions much faster than regular payments.

1.2 Problem

Since digital platforms are everywhere, marketing strategies have also changed a lot in the last decade. In the past, the companies were focusing on promoting their products only on TVs or on billboards. However, due to the digital era, social media platforms, and social media users have increased tremendously. Now there are new ways to reach the consumers who spend most of their time behind screens. One of the most modern marketing methods is Influencer Marketing, where the brands use micro-influencers and macro-influencers in social media to promote their products. Instead of creating traditional advertisements, they reach potential consumers with the help of social media influencers. In this context, Blockchain technology can help to build trustworthy and transparent platforms between brands and influencers and between influencers and their fans. By using Blockchain technology, brands can reduce their budget and time spent on traditional advertisement, communication, and payment processing. Most of the time, social media influencers have an ”ideal” body, affecting how the product they are advertising looks on them. However, since fans have different body shapes and what they might like on an Instagram influencer might not look the same on them. It leads to problems in online shopping. Since clothes look different on the buyers, it results in increased returns in online shops, which is a major problem for retailers [47]. In the existing applications, macro-influencers' fans or micro-influencers are not involved in the campaigns brands have created. The proposed framework brings a possibility for micro-influencers also to participate in the sub-campaigns that macro-influencers create based on the campaigns that brands have initiated. In this way, on Instagram, micro-influencers also share their pictures wearing the same outfits described in the campaigns. Seeing the same outfit on someone rather than people with an ideal body might give potential buyers better idea of how the outfit would look on them. In this master thesis, we will review the general state of the art on applications of Blockchain technologies in the Fashion and develop and propose a framework in Blockchain to handle the mentioned problems.

1.3 Purpose

The purpose of this thesis is to identify the sizing problem which causes the increased rate of returns in online shops and to study the possible ways to overcome that. By dragging micro-influencers into the process of promoting products of brands, this purpose will be achieved. Since there are many micro-influencers in social media, instead of giving them the possibility of direct communication with brands, macro-influencers can play a role as a bridge between them. The described architecture of the proposed solution can be beneficial to the readers who want to implement smart contracts on the blockchain, together with different back-end and front-end technologies.

1.4 Goal

The main goal of this degree project is to develop a distributed Blockchain framework that can be used to enhance the role of digital influencers in advertising for brands and minimizing the number of returns in online shops. The subgoals of the project are as follows:

- Reviewing state of the art on Blockchain technologies in eCommerce applications.
- Proposing a framework tailored for the increased rate of returns in online shops due to the sizing problem as an extension to the currently available social influencers Blockchain services.
- Reviewing the Blockchain architecture.
- Exploring the feasible technical methods to implement the framework.
- Developing User Interface to present the proposed solution.
- Experimenting with the proposed framework by analyzing and evaluating under the context of Fashion.

The deliverables will be the thesis report and the UI presentation of the framework proposed.

1.5 Methodology

The qualitative research methodology is the process of analyzing non-numerical data, whereas the quantitive research methodology is formed from a deductive approach and is about analyzing numerical data. In this thesis, in order to use techniques of both, a mix of qualitative and quantitive research methods is used.

1.6 Delimitations

The proposed model is deployed, tested, and analyzed in the Ropsten [1] blockchain network, which runs similar algorithms as the main Ethereum network [29]. Due to transaction costs, the main Blockchain network is not used for the deployment of smart contracts.

1.7 Sustainability and Ethics

Since the framework proposed gives potential buyers a better idea of how clothes would fit on them before proceeding with buying them, it will result in decreased returns in online shopping. The sustainability concern should appear before a consumer decides the purchase. After the purchase decision is made, it isn't easy to have a solution from the sustainability aspect [23]. According to the research conducted, there is not a huge difference in what return method (returning to a physical store, parcel delivery, etc.) is used. All of them have a bad impact on the environment in terms of CO2 emission [27]. By impacting the behavior of consumers with the help of micro-influencers, decreased returns is going to have huge benefits on sustainability.

When conducting the research, no participants or user data were involved. Technical tools used during the implementation are open for public use. All the results and ideas that appeared throughout the research is included in the thesis report with openness.

[1] Free of cost Blockchain testing environment for development.https://ropsten.etherscan.io/

1.8 Outline

The structure of the report is outlined as follows. Chapter 2 provides background information about Blockchain, Blockchain architecture, Smart Contracts, and Influencer Marketing necessary to follow the rest of the report. Chapter 3 gives a theoretical description of the methodology used in the study. Chapter 4 describes the system design of the proposed model and its implementation. In Chapter 5, the results and analysis are presented. Chapter 6 summarises the conclusions of the report and describes possible future work.

Chapter 2

Background and Related Work

A blockchain is a decentralized ledger of all transactions across a peer-to-peer network that enables peer to peer transfer of digital assets without intermediaries. Blockchain as a technical word was invented by an unknown author and inventor of Bitcoin, Satoshi Nakamoto, in the white paper [30] of the first cryptocurrency, Bitcoin. Bitcoin is the most famous example that is based on Blockchain technology. Nowadays, Blockchain is beyond bitcoin since the application of Blockchain is not limited to only the financial sector. The definition of Blockchain suggested by Melanie Swan in her book [39] is expressing not only what is Blockchain, but also the power of it: "We should think about the Blockchain as another class of thing like the Internet—a comprehensive information technology with tiered technical levels and multiple classes of applications for any form of asset registry, inventory, and exchange, including every area of finance, economics, and money; hard assets (physical property, homes, cars); and intangible assets (votes, ideas, reputation, intention, health data, information, etc.). But the blockchain concept is even more; it is a new organizing paradigm for the discovery, valuation, and transfer of all quanta (discrete units) of anything, and potentially for the coordination of all human activity at a much larger scale than has been possible before". [39] Currently, the digital economy is based on certain trusted authorities (e.g., a central bank). People use digital financial services by relying on those trusted parties for the security and privacy of their digital assets. The application of Blockchain in the financial sector offers many advantages over traditional banking:

- Transparency: the entire chain of the flow can be audited by anyone, which is why

Blockchain is a decentralized distributed ledger. That is due to the ability of easily getting access to historical transactions, as it provides much more transparency than traditional banking.

- Lower transaction costs: using the distributed ledger approach to form a system that anyone can use for peer-to-peer transactions, without authentication of the central agency, the Blockchain helps significantly in reducing the development and operation costs [50]. Since there is no central agency, there are no charges applied by third parties for a transaction, so the transaction fee is much lower than it is in traditional banking.
- Faster transactions: the fact that the transaction does not go through third parties will lead not only to lower transaction fees but also to faster transactions.

There are three types of blockchains, depending on the accessibility of the chains. **Public blockchains** are open-source, where anyone can send transactions as a user, or they can be miners. Bitcoin, Ethereum are examples of public Blockchain. On the other hand, **private blockchains** are more restrictive, and they are usually used by organizations. Thus, private blockchains are more centralized compared to public blockchains. The organization that owns the private blockchain handles write operations. However, the read permissions can be public or restricted to some participants [17]. **Consortium blockchains** are the specific type of private Blockchain. The main difference is that the consortium blockchain is managed by a group of entities instead of a single organization in the case of private Blockchain. Since consortium blockchains let different entities to control the chain, it helps multiple organizations to collaborate on the business process [11]. Hybrid blockchains are the combination of both private and public Blockchain. In Hybrid blockchain systems, any node can participate in the consensus process. However, generating the next block is only permitted for only some nodes [18].

2.1 Blockchain Architecture

Blockchain is a sequence of blocks that are chained together to store information related to transactions that occur on a blockchain network. A block consists of the block header and the block body [48]. A block header is used to identify a particular block on the Blockchain, and it has several fields:

1. Version: the block version number is four bytes that indicate which set of block validation rules to follow [8].

2. Time: a Unix epoch time when the miner started hashing the header [8].

3. Nonce: a 4-byte number, which is the result of the process of the mining. Nonce is the number used to generate the block, which is added to the Blockchain. Miners compete to find a nonce. It is manipulated by the publishing node to solve the hash puzzle [45]. If a miner finds such a nonce, they receive the block reward.

4. Merkle root hash: it is derived from the hashes of all transactions included in the block, ensuring the quick verification of blockchain data [8].

5. Hash value of the previous block: it secures the chain. It is one of the most important concepts in Blockchain. In order to change these values, all the previous hash values need to be changed. (see Figure 2.1.1)

6. nBits: it is referred as difficulty target and it is composed of 4 bytes. Difficulty target an encoded version of the target threshold this block's header hash must be less than or equal to [8].

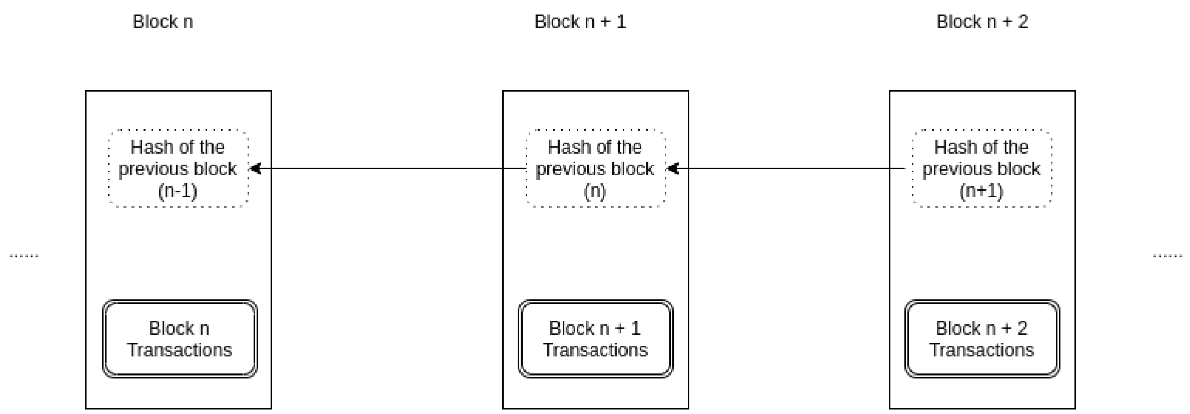

Figure 2.1.1: Each block has a hash of the previous block, the previous block is a parent block for every block

2.1.1 The Consensus Algorithms

Blockchain uses the decentralized consensus mechanism in order to achieve reliability in the blockchain network. There are several consensus algorithms used in Blockchain, mainly, PoW (Proof of Work), PoS (Proof of Stake), DPoS (Delegated Proof of Stake), and PBFT (Practical Byzantine Fault Tolerance) [28].

Double-spending is one of the main problems with digital transactions. Double-spending means that the owner spends the same coins again. Bitcoin solved the double-spending issue, which was a problem until blockchain cryptography. It was a problem for all kinds of digital assets since it was possible to copy and spend them more than once. The core difficulty was that there was no way to confirm the double-spending without a third party [39]. So, a common solution to this problem is to have a central authority that controls it. However, this solution would lead to a system like traditional banking. The solution described in the white paper of Bitcoin was to implement the proof-of-work [30], which is a mathematical puzzle to find a value, called a nonce, that produces the next block broadcast to the network. Proof-of-work is a consensus algorithm. Consensus algorithms are at the core of the Blockchain architecture which are mechanisms that make all blockchain nodes have a common agreement, and they ensure that the latest block has been added to the chain [25]. The presence of the consensus protocol in Blockchain also plays an important role in its security. The core idea of proof-of-work is to allocate rewards through the hashing power competition among the nodes. The first node that solves the mathematical problem creates the next block and gets a reward [28]. The nodes that try to find the hashes are called miners, and the overall process of proof-of-work is called mining. There can be a case that two miners find a solution simultaneously, which leads the chain to diverge into two potential paths. It is called a fork. To resolve forks, the protocol defines how miners should choose the chain to mine and add a block on. The criterion is the chain, which requires the most mining power to generate, is the winner. It means that the longest chain (see Figure 2.1.2) is the next path to continue.

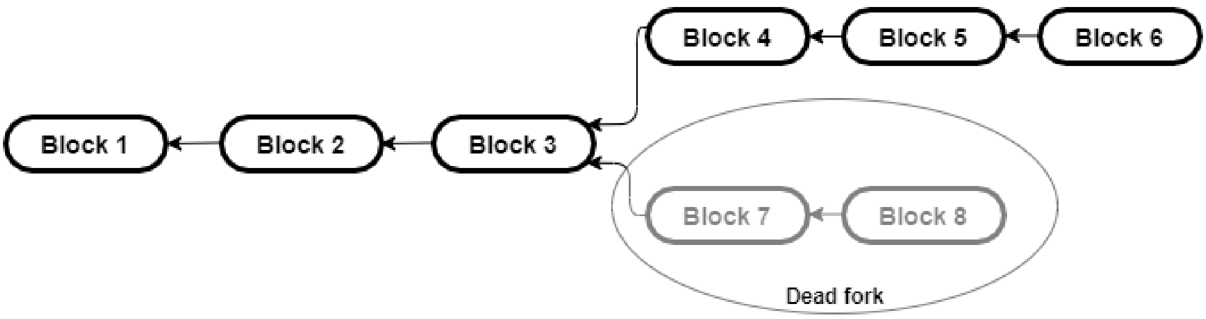

Figure 2.1.2: A fork happens, since the upper branch is longer, it is the winner.

So in order to be able to control the blockchain block generation, the entity should control at least half of the world's hashing power to become the first one to generate the latest block and master the longest chain. So the cost of achieving this is much more than the benefit. [28].

Most of the existing blockchain systems use the PoW mechanism to have a consensus. However, since mining requires a lot of energy in the PoW mechanism, the Proof of Stake algorithm is one of the alternatives to Proof of Work. In Proof of Stake, each coin's age is taken into consideration. When the coin is spent in any transaction, its age becomes zero. While in PoW, the longest chain is the winner, the Proof of Stake system uses a chain with the highest consumed coin age. [6] Peercoin uses the coin age method, so in Peercoin, the mining of the next block depends on whoever has the old and larger set of coins. So Compared to PoW, in PoS, being selected node has nothing to do with the computing resources of the node. That is why PoS saves more energy and is more effective [48] . TPS (transaction per second) is also higher in PoS than PoW because the PoS relies on the stakes that nodes own, not on the computing power [24]. As a result, the transactions are processed faster, which means higher TPS.

2.2 Applications of Blockchain

Blockchain is not only about Bitcoin. Nowadays, blockchain applications have been implemented in several areas.

Some intuitions and universities have used blockchain technology in solving problems in education. One example can be saving the credits, transcripts of grades, and certificates in the Blockchain. One of the universities that have already implemented this is The University of Nicosia. It is the first school that uses blockchain technology to manage students' certificates [37]. A project by the Media Lab Learning Initiative and Learning Machine can be another example of the applications of the Blockchain in education. It is an ecosystem for creating, sharing, and verifying blockchain-based educational certificates. All digital certificates are cryptographically signed and registered on the Bitcoin blockchain [13].

There are some existing blockchain applications in the medical area also. Most of the applications are usually about handling the patients' records. For example "MedRec" project uses blockchain technology to share records, achieving confidentiality and integrity. It also provides patients detailed information about their records and allows access not only to the patients themselves but also to the various providers and treatment institutions. [5]

Blockchain technology is also being applied to the insurance industry. One of the examples of the blockchain application in the insurance industry is InsureX, which aims to solve the challenges in the traditional insurance market [10]. According to the report conducted by Deloitte [9], there are several pain points in traditional insurance which blockchain implementation is solving: e.g., use of middleman in the process, insufficient exchange of information, being fraud-prone, manual claims review and processing, etc.

Blockchain applications are also bringing new possibilities for protecting the rights of the different entities, especially in the Internet. One example can be file copyright. If we consider that one file which is copyright protected can be easily duplicated and copy several times across the network. However, the implementation of Blockchain in this area makes some of these challenges solvable. Moreover, the fact that all the changes added to the distributed ledger can never be modified makes the file owner's information trustworthy, and easily verifiable [10].

2.3 Ethereum

Ethereum is a distributed blockchain network for decentralized networks and smart contracts. Ethereum was introduced by Vitalik Buterin, and it has some advantages over Bitcoin technology. The main difference between Ethereum and Bitcoin is being full turing-completeness. It means that Ethereum supports all types of computations. Ethereum blockchains can be applied not only in financial sector, but also in different kinds of concepts. Smart contracts are lines of code which are executed when some terms and conditions are met. More details about smart contracts will be explained in the next section. The currency for Ethereum is called Ether. As it is stated in Ethereum whitepaper [1], "Ether is the main internal crypto-fuel of Ethereum, and is used to pay transaction fees". *Wei* is the smallest unit of Ether. An ether is 10^{18} Wei. Except for Wei, there are also other different subdenominations of Ether (see table 2.3.1).

With the help of smart contracts, any blockchain developer can create their terms and conditions valid for their business flows and deploy them to the public ethereum network. So with smart contracts, Ethereum enables everyone to have their own rules for some certain flows [42].

Two types of accounts are possible in the Ethereum network. One of them is

Multiplier	Name
10^0	Wei
10^{12}	Szabo
10^{15}	Finney
10^{18}	Ether

Table 2.3.1: Different subdenominations of Ether (reprinted from the paper of Ethereum: A Secure Decentralised Generalised Transaction Ledger [14]).

externally owned accounts, which are controlled by private keys. The other one is contract accounts. In contrast to the externally owned accounts, contract accounts are controlled by their contract code [1].

In Ethereum, accounts represent states. They consist of 4 components [7]:

1. nonce: it is assigned if the account is a contract account.It represents the number of contracts created by that account. Otherwise, if the account is an externally owned account, then nonce is the number of transactions sent from the account.
2. balance: this field represents the amount of Wei the account has.
3. storage root: a 256-bit hash of the root node of Merkle Particia tree. The tree encodes the storage contents of the account. By default, it is empty.
4. code hash: this component represents the hash of the EVM (Ethereum Virtual Machine) code of the account. Code hashes are executed when the account address receives a message call. They are immutable.

Blocks consist of transactions, and all the blocks are connected by a cryptographic hash. We can think of blocks as a journal or a ledger. It records a series of transactions, and it also contains an identifier for the final state [14] . All the blocks are chained together with the cryptographic hash functions.

There are two types of transactions in Ethereum. One of them is the transactions resulting in message calls, and the other one is the transactions sent to interact with the smart contract.

Both transaction types have a common data structure and contain common data fields [14]:

1. nonce: a value equal to the number of transactions sent by the sender. Each time a sender sends a new transaction, the nonce will be increased by one.

2. gasLimit: the maximum number of gas that can be paid to perform the transaction.

3. to: the address of the recipient whom the message call addressed to.

4. value: a value to be transferred to the recipient as a result of a transaction.

5. v, r, s: values which define the signature of the message call and they are used to determine the sender.

In the Ethereum network, there is a specific measurement to measure the computational work for running each transaction, which is called Gas. Gas is a control mechanism for the resources that a transaction can consume. In the Ethereum network, when the user creates a new transaction, there is a fee paid by the user for the network to execute the transaction. Some parts of this fee, which is called Gas as mentioned above, is for computational and also storage costs of running a transaction [46]. The amount of gas involved in transaction processing also helps preventing denial-of-service attacks. Gas price is another term used when talking about this mechanism in the Ethereum network. It only represents the number of Wei per unit of gas.

As the name suggests, the gas limit is the limit or maximum price the user is willing to pay when creating a new transaction or even performing a function of a smart contract. We can think of the gas limit as a maximum budget for the miner in the Ethereum network to process the transaction. When the miner performs any computational operation, the gas is subtracted from that budget. In case the paid Gas is not enough for executing the whole transaction, then an Out-of-Gas exception will be thrown by the EVM. However, it will still be included in the chain. Otherwise, when the paid gas is more than needed to execute the transaction, then the rest of the gas will be refunded to the user [46]. Miners who verify and process a transaction are taking this fee for the work they have done. By setting a threshold, the miners can choose not to handle the transactions if the fee is low. A higher gas fee means that the transaction will be completed faster than with lower gas fees. The total cost the user is willing to pay for performing a new transaction is the multiplication of gas limit and gas price.

$$totalCostOfTransaction = gasPrice * gasLimit$$

2.4 Smart Contracts

A smart contract is a program that runs on the Ethereum blockchain. It goes beyond the transfer of the cryptocurrency assets. It can be applied to many fields and applications. A smart contract was first mentioned in the paper [40] by Nick Szabo. Nick Szabo suggested that the synthetic assets might be applied in smart contracts: ”Another area that might be considered in smart contract terms is synthetic assets. These new securities are formed by combining securities and derivatives in a wide variety of ways. Very complex term structures for payments can now be built into standardized contracts and traded with low transaction costs, due to computerized analysis of these complex term structures.”

Some variations of smart contracts exist in the blockchain context. The most well-known and worldwide used implementation is in Ethereum. Smart contracts are business rules defined by some lines of code, which are automatically executed when the predefined conditions are met.

The benefits of smart contracts can be summarised as follows:

- They simplify the complex process or business flows, which involve many middle intermediaries because of a lack of trust among participants in the transaction [16]. So they can be executed without any human intervention.
- The financial settlement will be automatically executed once the predefined condition is met. As a result, the turnaround time can be significantly reduced [49].
- Smart contracts cannot be altered compared to traditional contracts. So once it is created and deployed, every party who has been involved in the smart contract is certain that when predefined conditions are met, the smart contract will be executed.
- Smart contracts are never lost. They are always on the blockchain. On the other hand, traditional contracts can be lost.
- Smart contracts decrease transaction costs. Unlike regular contracts, in smart contracts, there is no cost associated with ensuring contractual performance (e.g., litigation costs or costs associated with the provision of collaterals) [36].

However, they have the following disadvantages:

- Although smart contracts reduce the cost compared to traditional contracts, the infrastructure necessary for the implementation of smart contracts and costs associated with the development is still high [36].
- Immutability of the smart contracts can be considered as a disadvantage in some cases. For example, if there is a bug in the already deployed contract, it is impossible to alter the contract.
- Legal status of smart contracts is still not certain.

Smart contract codes run on the EVM (Ethereum Virtual Machine). Anyone with the necessary skills can write a smart contract and deploy it to the network. Smart contracts can be implemented using high-level programming languages such as Solidity, C++, Java, JavaScript, etc. Solidity is the main and widely used smart contract programming language. Every smart contract in the blockchain has a unique contract address, and they hold a virtual coin called Ether, as mentioned in the section above. [26] A smart contract is triggered by sending a transaction to it. A transaction includes payment for that transaction and data if it is necessary for the contract function. For processing each transaction to smart contracts, there is a fee to be paid to the miners to execute the smart contract. This fee is subtracted from the multiplication of the gas price and gas limit specified by the user who is sending the transaction.

An Ethereum smart contract holds a state consisting of:

- a private storage
- the amount of virtual coins (Ether) [26]

They are immutable. Once they are created, it is impossible to change the code of a contract. The smart contract is not like a traditional software application code that can be modified and deployed on the same location or address. When a smart contract is changed to be deployed again, it will be deployed as a completely new smart contract. That is why smart contracts are considered immutable.

2.5 Influencer Marketing With Blockchain

Since the usage of social media is increasing, social media marketing is now one of the most effective ways of marketing. According to the Digital 2019 [12] report provided by "We are Social" agency, 57 percent (4.388 billion) of the world's population is connected to the Internet. Approximately 80 percent of the people connected to the Internet are active social media users. The amount of time spent online per user is also significant, with an average of more than 6.5 hours per day. These statistics encourage brands and companies to use social media marketing methods more than traditional marketing. In the digital era, there are different approaches to reach many potential consumers in an effective way. One of the most effective methods is to follow some recommendations from influencers. Influencers are popular social media users who proved to have influential power on customers' purchase decisions. By using that power, they can advertise the products of the brands and get real engagements on the content. Nowadays, the marketing of products is a real challenge due to competitive producers and sellers in the game [32] . In traditional marketing, word-of-mouth advertising is very important for the business. We can say that Influencer Marketing is also one way of word-of-mouth advertising. The significant amount of Internet usage has extended potential consumers' options to reach unbiased product information from other consumers. It also provides the opportunity for consumers to offer their consumption related advice about products in electronic word-of-mouth (eWOM) [19].

Influencer Marketing can be more effective if it is integrated with Blockchain technology. With blockchain application, dApps (Decentralized applications) can be implemented, which are transparent, distributed, flexible, and open source. Due to being open-source, the dApp is gaining the trust of potential users, so if the app is open source, it brings the transparency needed to improve the application with the help of other developer's contributions [33]. A dApp is not owned by any node in the network. With the application of smart contracts on the blockchain dApp, more trust and transparency is brought to the many possible application areas, as well as Influencer Marketing.

Adopting blockchain dApp with influencer marketing brings several advantages to the brands and the influencers which are summarised as follows:

- Higher level of trust: since blockchain dApp provides more transparency, trust is easily established between influencers and brands.
- Easier processing of payments: with the help of smart contracts and cryptocurrencies, after certain preconditions are met, the influencers are paid instantly.

- Faster process and faster communication: no paper contracts are needed, and in addition to that, influencers don't need to spend their time negotiating with different brands and companies. Instead, they manage everything in one platform. Brand investors and companies are also involved in faster communication with investors with third parties and contract signing.

2.6 Related Work

There are already existing Blockchain platforms that involve social media influencers. In this section, we highlight some examples of these platforms.

2.6.1 IndaHash

IndaHash [21] is one of the leading platforms that connect brands with influencers on social media. It was launched in 2016. IndaHash has its a web-based solution and mobile application. IndaHash automates the process of negotiation, where brands create a task by setting the budget and the campaign details like conditions to complete the assignment (hashtags, content to be posted, etc.). By applying the created tasks the influencers join the campaign started by brands. After the influencers applied for the campaign, they publish content as required with necessary hashtags on social media platforms.

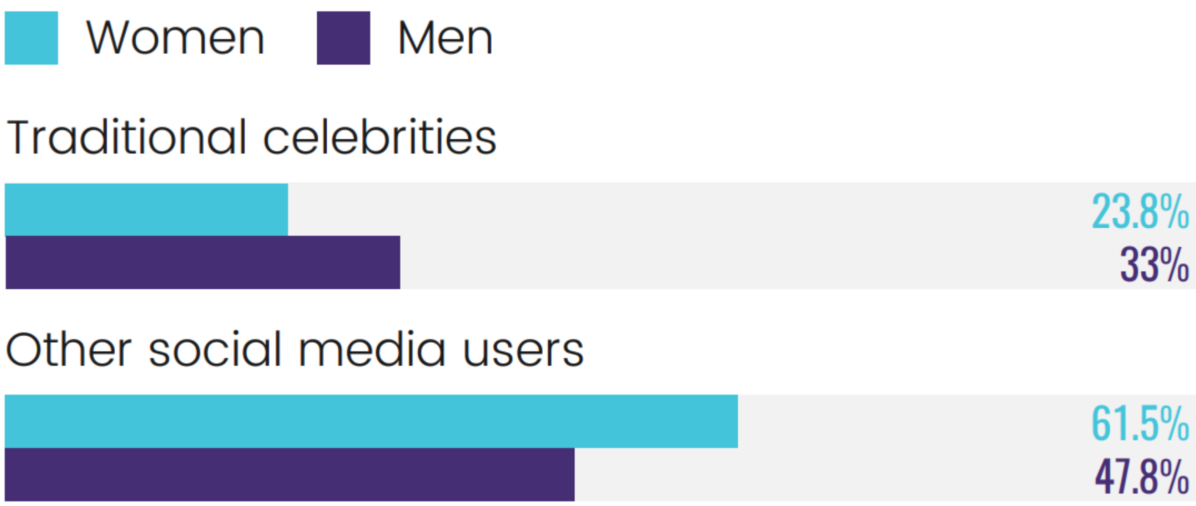

Figure 2.6.1: What or who is the most opinion-forming for you? (reprinted from the international study conducted by IndaHash Lab [44]).

IndaHash has conducted an international study [44] with over 2200 digital influencers in 2017, which brought excellent insight with statistics related to influencer marketing.

According to the results, 83 percent of the influencers bought a product they saw on social media last month. It shows the real power of social media marketing. It can sell even to influencers. The result in Figure 2 shows that female influencers trust social media users more than men do. In contrast, men trust traditional celebrities more.

As described in the white paper [22] of IndaHash, the business model is as below:

1. Influencers register on the platform.
2. The registration is approved after the manual inspection of the Instagram account.
3. Influencers apply to join the campaigns.
4. Influencers submit a proposal.
5. The proposal is checked by the moderator and the client.
6. After approval, the content is published.

2.6.2 APPICS

APPICS [4] is a Blockchain-based and reward-based social media platform that runs on top of Steem blockchain. Its aim is to reward people by turning likes into currency. With APPICS, the users of the platform can post their content and get the reward as likes on the posts. So, social media fans are rewarded for the time they spend on social networks. Unlike any other popular social media site, where whole platform's value and revenue goes to the shareholders, the APPICS platform enables all users to earn through the reward-token ”APX”. Since the possibility of getting many likes is getting more rewards, the content creators are more motivated to share posts that have better quality. It is also possible to send direct transactions to the other user's account without any additional fee or delay. The coin used in the platform is called APPICS coin (XAP). Later those earned XAPs can be top-up to the credit card. With the existing amount on the user's account, they can also buy the selected items. The active users can be voted by users to be influencers in one of the 15 categories like fashion, sports, etc.

2.6.3 PATRON

One of the existing Blockchain-based platforms that involve social media influencers is PATRON. PATRON is a fee-charging, Blockchain-based sharing economy platform that shares the value of influencers [31]. The aim of the project is to create a full ecosystem where Influencers and brands can connect directly. In this way, the PATRON project aims to replace the existing business model with a system that works better for both brands and social media influencers alike with fewer expenses without the involvement of any intermediaries [3]. With the help of the Patron platform, brands are able to book social media influencers without having to worry about a broker or other intermediaries [3]. PATRON platform uses PAT token wish is an ERC20 token implemented on the Ethereum Blockchain.

PATRON's concept is very similar to the Airbnb concept. While users on Airbnb share the room, On PATRON, influencers share their "social media post slot" with companies and brands as well as fans and followers [31]. So they are paid for their social media post slot, which a company books without an intermediary.

On the PATRON platform, Influencers are classified into four categories: Micro-Celebrity, Influencer, Micro-Influencer, Nano-Celebrity. Three categories are used in order to measure the level of the influencer: [31]

1. Power of Sending Message (Reachability)
2. Traction (Engagement)
3. Trustability(Evaluation by Third Party)

Anyone who falls under one of the categories can use the platform. As an example of the use case, we can take an influencer who is a singer and famous on Instagram. And on the platform, for that influencer, it is specified what type of slot post is available in the slot (e.g., live, status, or ordinary post sharing), as well as the price for that slot. The search functionality allows users to search for influencers around the world according to their needs. [31]

Chapter 3

Methodology

In this chapter, the research methodology and methods that are used in this thesis are described. Section 3.1 describes the research process. Section 3.2 describes the experimental setup for testing the proposed framework.

3.1 Research Process

In this research, in order to use different techniques of both qualitative and quantitive research methods, we follow mixed of both approaches.

Qualitative research is mainly about collecting non-numerical data and understanding the concept by analyzing the data. On the other hand, the quantitive research method involves collecting data in numerical form and analyzing the data with statistical calculations. The mixed research method is useful since quantitative data helps with the qualitative side of a study during design by finding a representative sample. In contrast, qualitative data help the quantitative side of the study during design by aiding with conceptual development [2].

3.1.1 Literature Review

As a first step of the research process, a quite extensive literature review was performed. Initially, blockchain domain knowledge was acquired. In the next phase of the literature review, the existing similar Blockchain applications was studied. After the literature review, the knowledge about the architecture of the Ethereum blockchain network and dApp implementation on the Ethereum blockchain was grasped. Since the Blockchain domain requires extensive technical knowledge to develop applications, that is also done as a part of this phase.

3.1.2 Model Proposal

In the next phase, the possible methods are investigated in order to overcome the problem of increased returns in online shops due to fitting problem of the advertised outfits. This step was finalized by proposing the framework for not only brands but also macro-influencers to create their own campaigns, which are available for their fans or micro-influencers to participate.

3.1.3 Smart Contract

Once the proposed solution was finalized, the smart contract is designed. In this step, necessary information to create campaigns is discussed. After the design process, the implementation of the smart contract is performed in Remix, which is a web-based compiler and IDE [34]. It is easier to build and debug smart contracts in Remix since it has built-in features available to use for development.

3.1.4 Implementation

After the smart contract was designed and implemented, the web platform was implemented to present the proposed solution. The web implementation includes User Interface and REST API development for simulating Instagram's basic functionalities necessary for our proposed solution. REST API is implemented in the Spring framework using Hibernate and Java. The UI is implemented using next.js. As the last step of this phase, web3js is used to connect the web platform to the local blockchain network which is available through Ethereum Ganache.

3.1.5 Deliverables

The deliverables are the thesis report and the simple UI implementation of how the proposed framework is working.

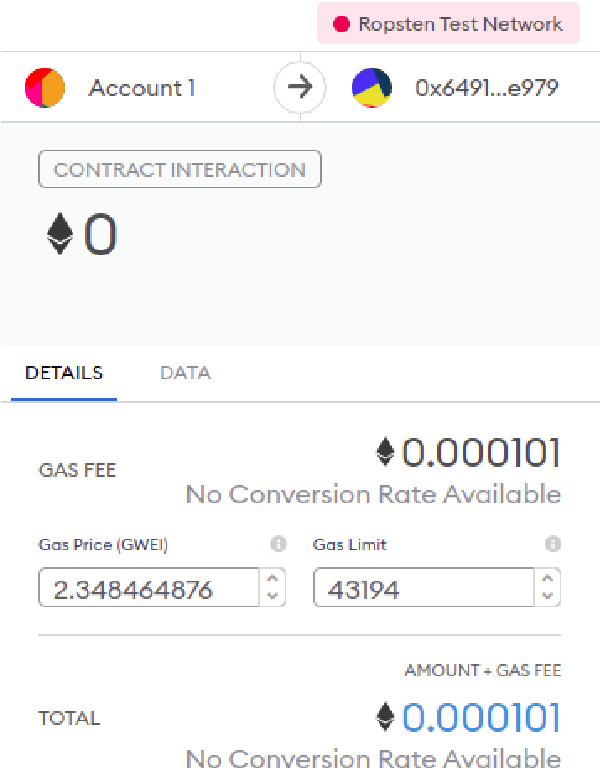

Figure 3.1.1: The notification in Metamask shows the gas price, gas limit and gas fee for executing the transaction in Ropsten test network.

3.2 Experimental Setup

There are several test networks available like Ropsten, Rinkeby, Kovan to test blockchain applications. To evaluate the performance of the proposed smart contract solution, the smart contract was deployed to Ropsten Test Network. Ropsten Test Network is a testing environment for smart contracts. It is one of the test networks that is most similar to the main network in Ethereum. Both the main Ethereum network and Ropsten Test Network use very similar consensus algorithm. Ropsten Network is free, which makes it suitable for our evaluations since no financial cost is involved in deploying smart contracts in Ropsten. Due to the financial costs, the evaluation of the proposed solution is performed only in the test network instead of the main Ethereum network. Metamask is a browser extension which allows to access the local and also test blockchain networks, including Ropsten. With the help of Metamask, a few Ethereum accounts were created in the Ropsten test network. After having some accounts in the test network, using browser-based Remix IDE, the smart contract of the proposed solution is deployed on the Ropsten Test Network. With this experimental setup, we manage to call the smart contract functions through Remix. And in the

Metamask, we can see the price for the execution of the specific transactions (see Figure 3.1.1). In order to confirm the successful execution of the transactions (see Figure 3.2.1) Etherscan (Ropsten Testnet Explorer) [35] is used.

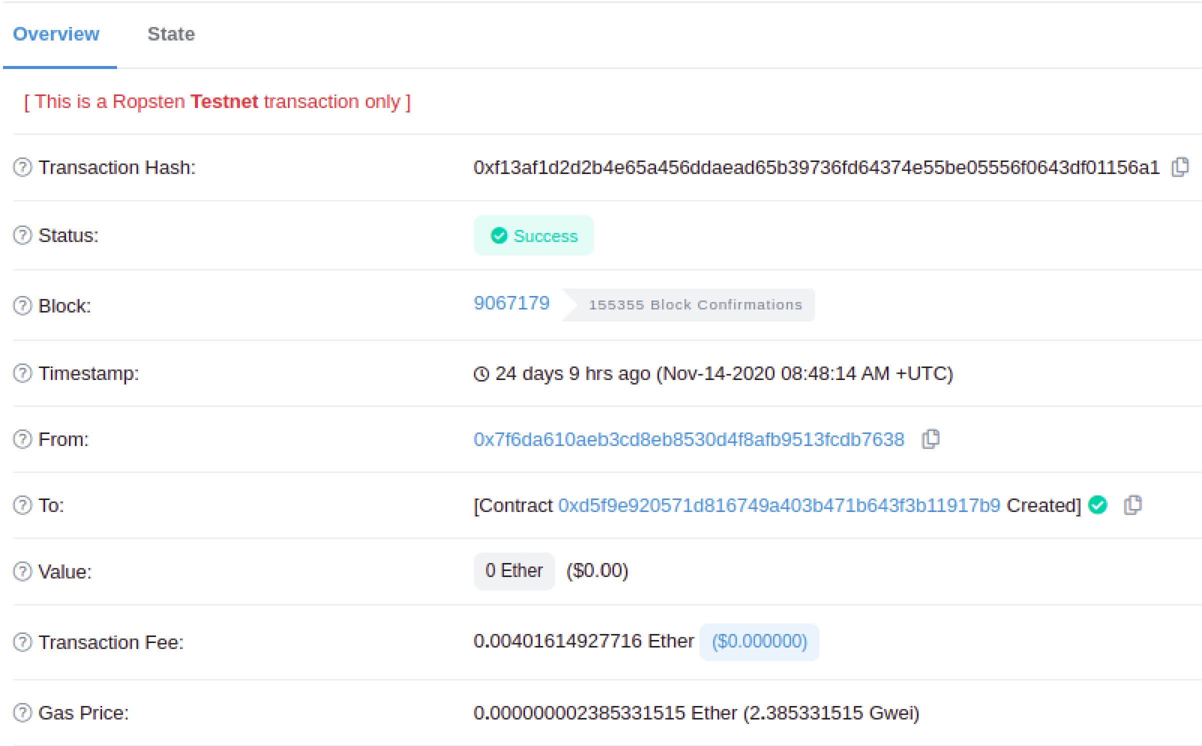

Figure 3.2.1: The transaction details of smart contract creation in Etherscan.

Chapter 4

Design and Implementation

This chapter describes the design and implementation of the project. System design is described in detail. Moreover, the interactions among the different components of the system are also explained in this section.

4.1 Technologies and tools

The proposed model consists of several components, which involve different technologies in the development.

Truffle: a development environment that provides a command-line tool to build, compile, and deploy Ethereum dApps using EVM.

Web3.js: Web3.js is an Ethereum JavaScript API. It is a collection of libraries that has builtin functions to interact with an Ethereum node. Those builtin functions [43] let us perform necessary actions using an HTTP connection.

- **web3.eth.Contract**: this object makes it possible to interact with the smart contract in the Blockchain network. If the new contract should be created, then the smart contract's JSON interface will be given as a parameter. Web3 will convert the call into a low-level ABI call over RPC.

```
new web3.eth.Contract(contract.abi)
        .deploy({ data: contract.evm.bytecode})
        .send({ gas: gasPrice, from: accountAddress});

```

- **web3.eth.getAccounts**: this function returns the list of accounts.
- **web3.utils.fromWei**: this function converts any wei value into an ether value.
- **contract.methods.myMethod**: when the methods are simply called, it can never alter the state of the smart contract. In order to modify the smart contract state, the transaction should be sent instead of just calling the method.

MetaMask: an Ethereum wallet that can be used in development by installing a browser plugin.

Solidity: a high-level language used to implement smart contracts on different blockchain platforms. [38]

Ganache: it allows creating a local blockchain for development in Ethereum blockchain. It is being used for the development, testing, and deployment of the smart contract. [15]

React: Next.js is used to create a User Interface to allow the users to interact with the existing smart contracts or create new smart contracts on the Ethereum blockchain.

Remix: Remix IDE is used to develop smart contracts, debug and test them directly on the browser. It is an open-source tool written in JavaScript

4.2 Architecture

In this section, the architecture of the proposed framework is discussed.

4.2.1 High level overview

The system proposed is supposed to be using Instagram social media. In order to make the development process smoother, we have developed a simple REST API that will simulate simple actions of Instagram by exposing a few endpoints.

- **POST-/users/username/posts/**: this endpoint is used to simulate sharing a post on Instagram.
- **GET-users/username/posts/**: this endpoint is used to simulate fetching all the posts of a user.

Next.js is used for developing a user interface for the proposed model. In the model proposed, there is a high coupling between smart contract instance and Instagram simulation. Web3j is communicating to the Ethereum platform, and by Fetch API, we interact with REST API. As a result of those interactions, we decide if the user is allowed to perform the intended actions (e.g., applying for the campaign).

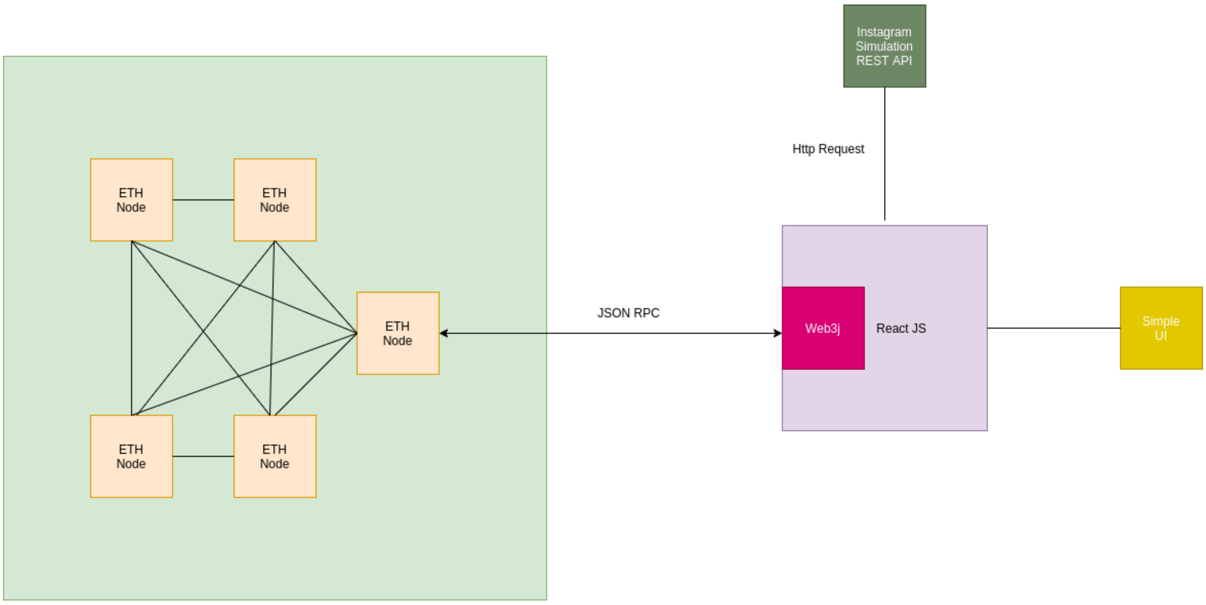

Figure 4.2.1: The high level view of the system design

4.2.2 System design

In the model proposed, there are three types of accounts can be created. **Brand account** is for brands and companies to publish their campaigns. **Macro-influencer account** is for the macro-influencers where they can apply and participate in the campaigns. **Micro-influencer account** type is for the fans of macro-influencers. Since the authorization and account management of the model proposed is out of scope, the Autho Authentication API [41] is used with the hard-coded account management implementation.

Two smart contracts are used in the model proposed. One of the smart contracts is called *CampaignCreaterFactory*, and the other one is Campaign. There are two types of campaigns that can be created (see Figure 4.2.3): **Campaign**, **SubCampaign** . The **Campaign** type can be created only by brand. On the other hand, **SubCampaign** type can be created only by macro-influencers. So the addressed audience for Campaign type is macro-influencers. Only they can apply for those campaigns. However, the SubCampaign type is targeted for micro-influencers.

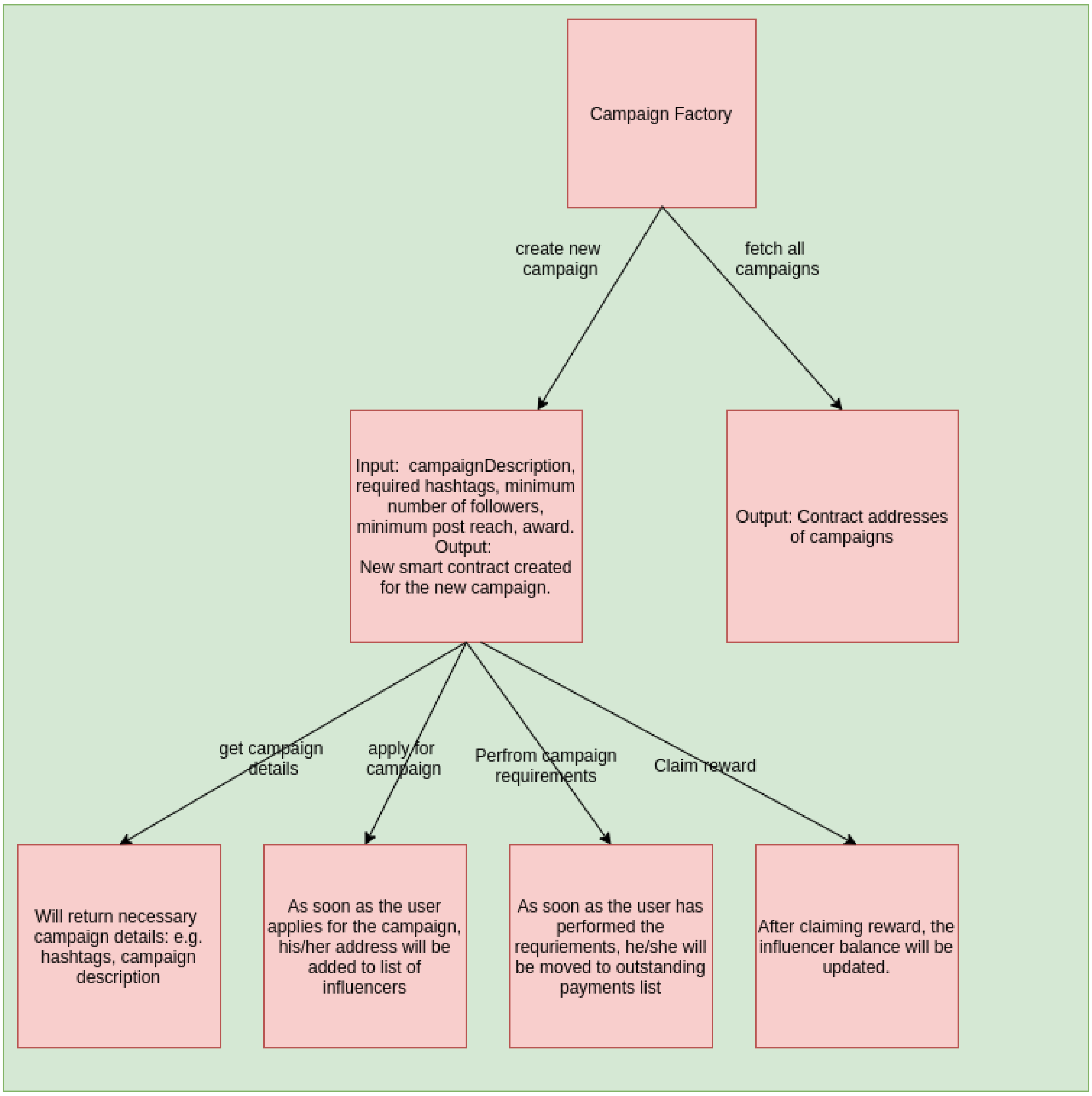

Figure 4.2.2: The high level view of the system design

The role of *CampaignCreaterFactory* is to be a factory for creating all other campaigns. By having *CampaignCreaterFactory*, we aim to collect all the campaigns in a factory, by which it will be easier to fetch all the existing campaigns.

```
contract CampaignCreaterFactory {
  address[] public deployedCampaigns;

  function createNewCampaign(uint minimumNoOfFollowers,
                             string memory addDescription,
                             string memory  requiredHashtags,
                             uint advertisementFee,
                             uint advertisementReach,
```

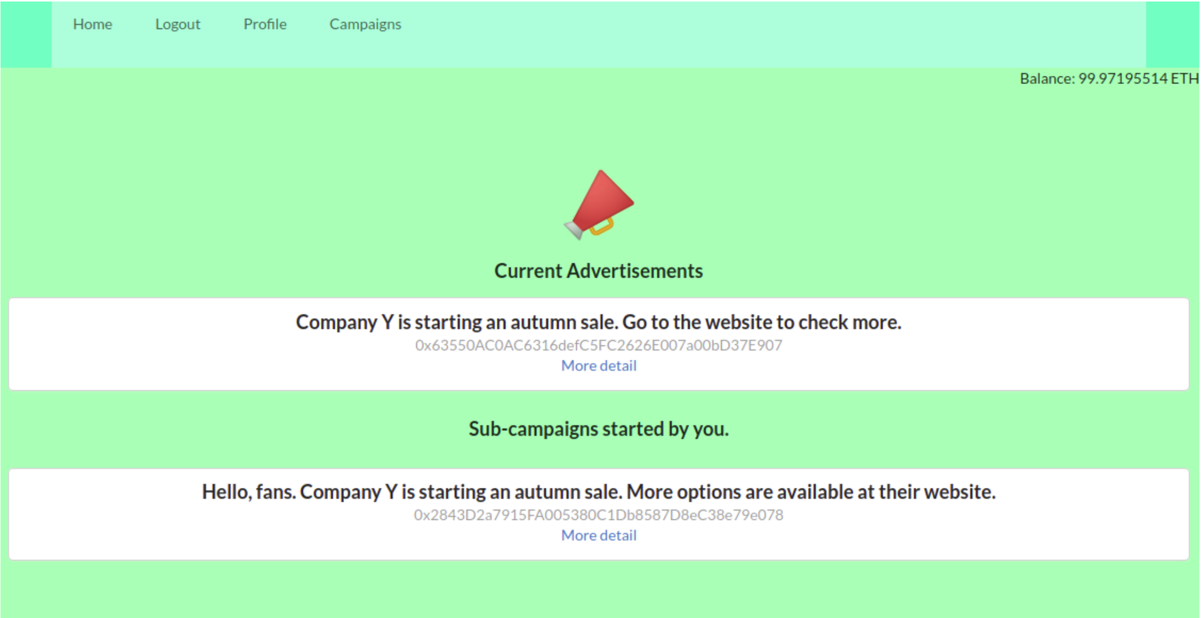

Figure 4.2.3: The page where the macro-influencers can view campaigns and sub-campaigns.

```
                             CampaignType newCampaignType)
                             public {
    address newCampaign = address(new Campaign(
                          minimumNoOfFollowers,
                          addDescription,
                          requiredHashtags,
                          advertisementFee,
                          msg.sender,
                          advertisementReach,
                          newCampaignType));
    deployedCampaigns.push(newCampaign);
  }



  function getAllDeployedCampaigns() public view returns
                                     (address[] memory) {
    return deployedCampaigns;
  }
}

```

The array of *deployedCampaigns* stores all the campaigns that have been created. They can be both *Campaign* type and *SubCampaign* type contracts. All the contract

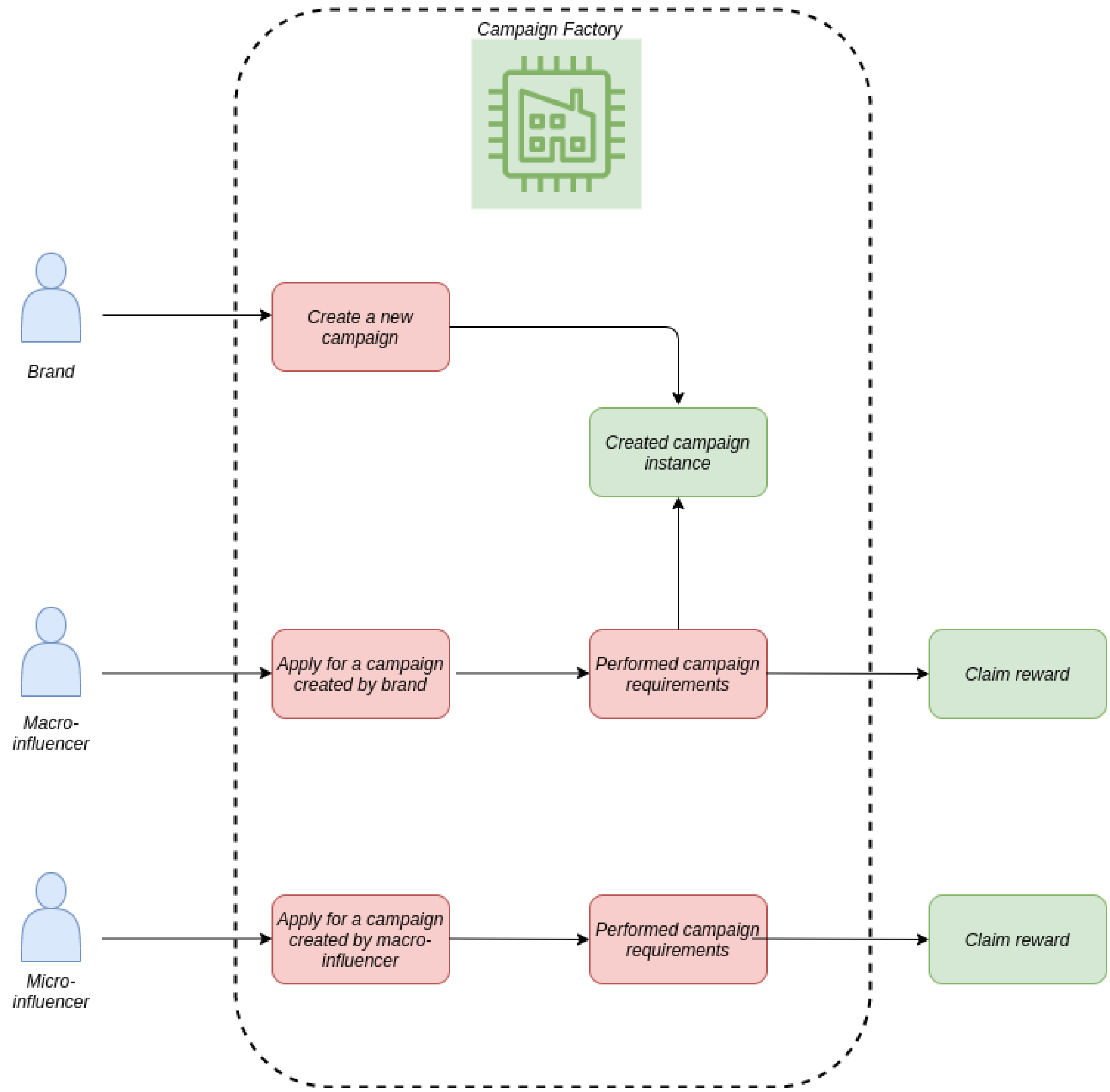

Figure 4.2.4: Summary of the interactions in the system

address of the campaigns are trackable due to storing them in the array. And the array type is address type since all it contains are the addresses of the contracts. We used an array over a map since, in Solidity, an array is more appropriate if the intended operation is iteration. The primary function which creates the campaign contract is *createNewCampaign*. It takes all the necessary parameters which come with the contract call. The parameters sent to the contract creation call is as below:

- **minimumNoOfFollowers**: the minimum number of followers of the influencer to be qualified to participate in the campaign.
- **addDescription**: set by the brand or by the macro-influencer, depending on who created the campaign.

- **requiredHashtags**: a field for the required hashtags of the campaign that the influencer must share with the post on social media.
- **advertisementFee**: the amount of reward the influencer will get in applying and achieving to reach the required number of audiences.
- **msg.sender**: this parameter is used to store the contract owner, the address of the entity who created the campaign.
- **advertisementReach**: the advertisement audience target (post engagement) to be reached to be allowed to claim the reward. Until the specified number of the audience has been reached, the influencers are not able to claim the reward.
- **newCampaignType**: the campaign type of *Campaign* or *SubCampaign* depending on who has created the campaign contract.

The other contract is the *Campaign* contract, which is called by the *CampaignCreaterFactory*. Once a brand clicks on *Create Advertisement* button (see Figure 4.2.5), the page for creating a new campaign is loaded. In that page, all the necessary details for a campaign is filled (see Figure 4.2.6) . Once a brand clicks on the button to create a new campaign, the *CampaignCreaterFactory* contract calls the constructor of the *Campaign* contract, which results in new campaign creation. In the Campaign contract, we have created an *Advertisement* struct that stores the necessary details of the campaign contract. In Solidity, a struct is a type that can be defined with any number of fields.

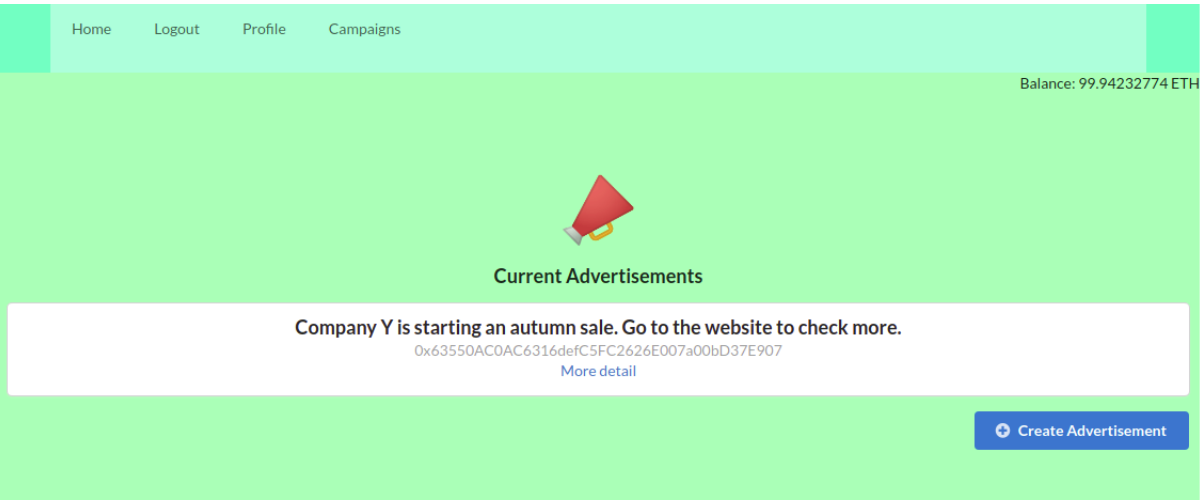

Figure 4.2.5: The page where brands can view the campaigns they have created. Once they click on *Create Advertisement*, they are routed to the page for creating a new campaign.

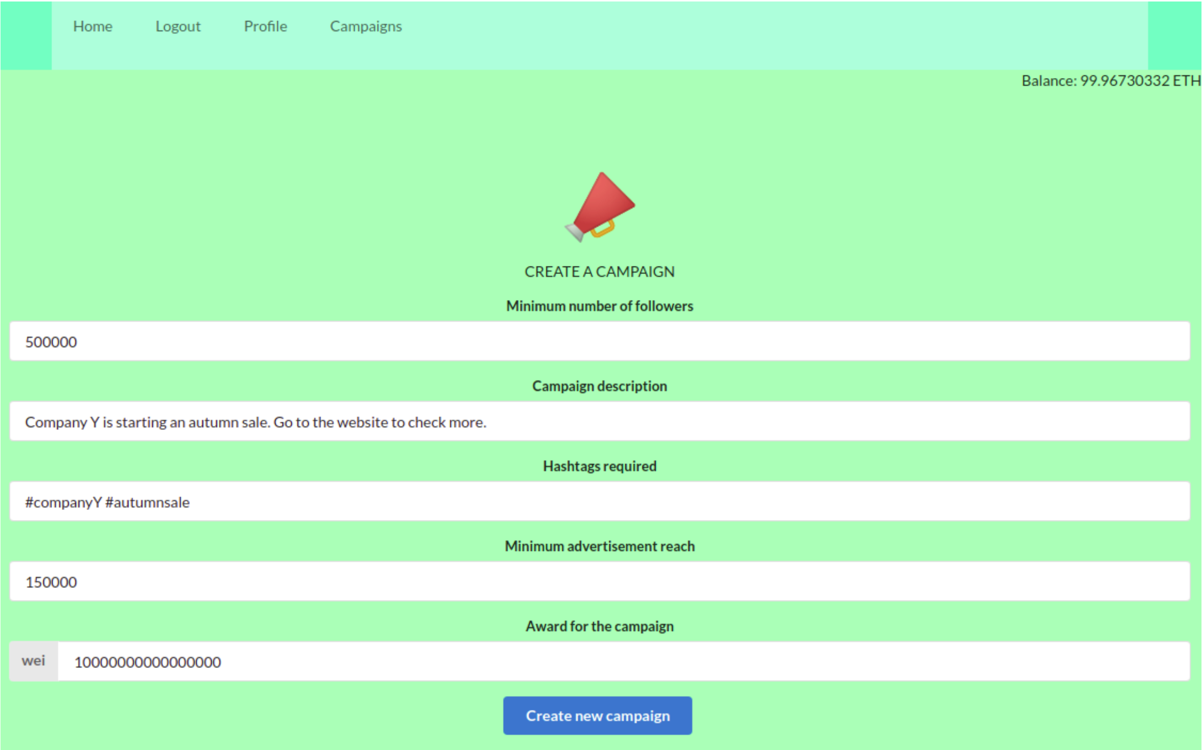

Figure 4.2.6: The page where brands create a new campaign.

```
struct Advertisement {
        string description;
        string hashtags;
        uint payment;
        uint addReach;
    }

```

The page to view the specific campaign details is almost the same for all three account types. The campaign details like the minimum number of followers, campaign description, required hashtags, minimum advertisement reach, and campaign award are displayed on the page. The only difference is that the buttons to apply, to perform requirements, and to claim the reward is only shown in case you are eligible for the campaign.

The most important thing to mention in Campaign smart contract is three maps that store addresses of the influencers.

```
    mapping(address => bool) private alreadyAppliedInfluencers;
    mapping(address => bool) private currentInfluencers;
    mapping(address => bool) private outstandingPayments;

```

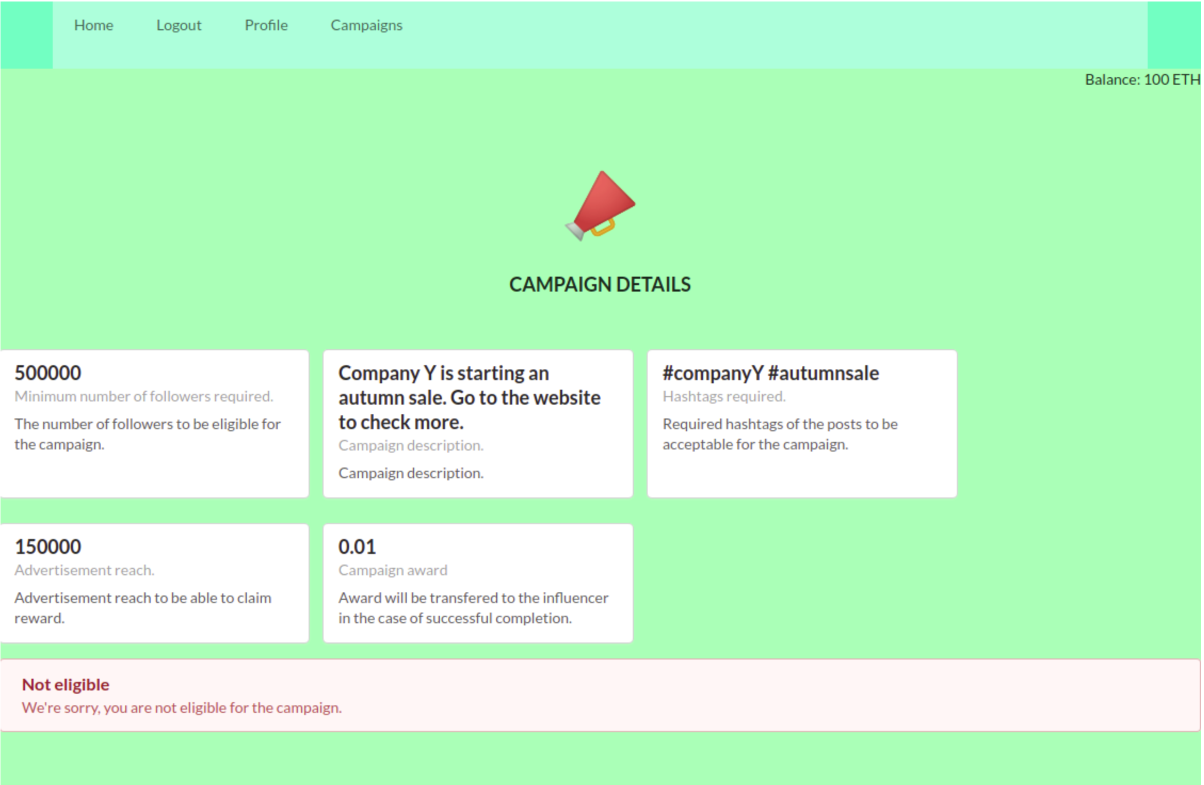

Figure 4.2.7: The information message shows that the macro-influencer is not eligible for the campaign.

- **alreadyAppliedInfluencers**: this map stores the account addresses of the influencers who have already participated in the campaign. Having this map is essential to keep track of the account addresses in order to prevent an influencer from participating in the campaign several times.
- **currentInfluencers**: this map stores the account addresses of the current influencers. Current influencers are the influencers who have applied for the campaign, but they have not yet performed the requirements of the campaign in order to be eligible to claim the award.
- **outstandingPayments**: this map stores the account addresses of the influencers who performed the requirements of the campaign, but they haven't yet claimed the award. As soon as the influencer claims the award, they are removed from this map and added to *alreadyAppliedInfluencers* map.

4.3 Implementation

In this section, the functions of the smart contract are described in detail.

4.3.1 Apply For Campaign

The primary role of having three maps is the security checks before the influencers apply for the campaign or claim the campaign award. When the influencer applies for the campaign, firstly, it is checked whether the contract address of the influencer exists in the *alreadyAppliedInfluencers* map. If it doesn't already exist in the *alreadyAppliedInfluencers* map, then applying for the campaign will be successful. In case the account address of the influencer exists in the map, applying for the campaign doesn't end successfully.

```
function applyForCampaign() public payable {
  require(!(alreadyAppliedInfluencers[msg.sender]));
  currentInfluencers[msg.sender] = true;
}

```

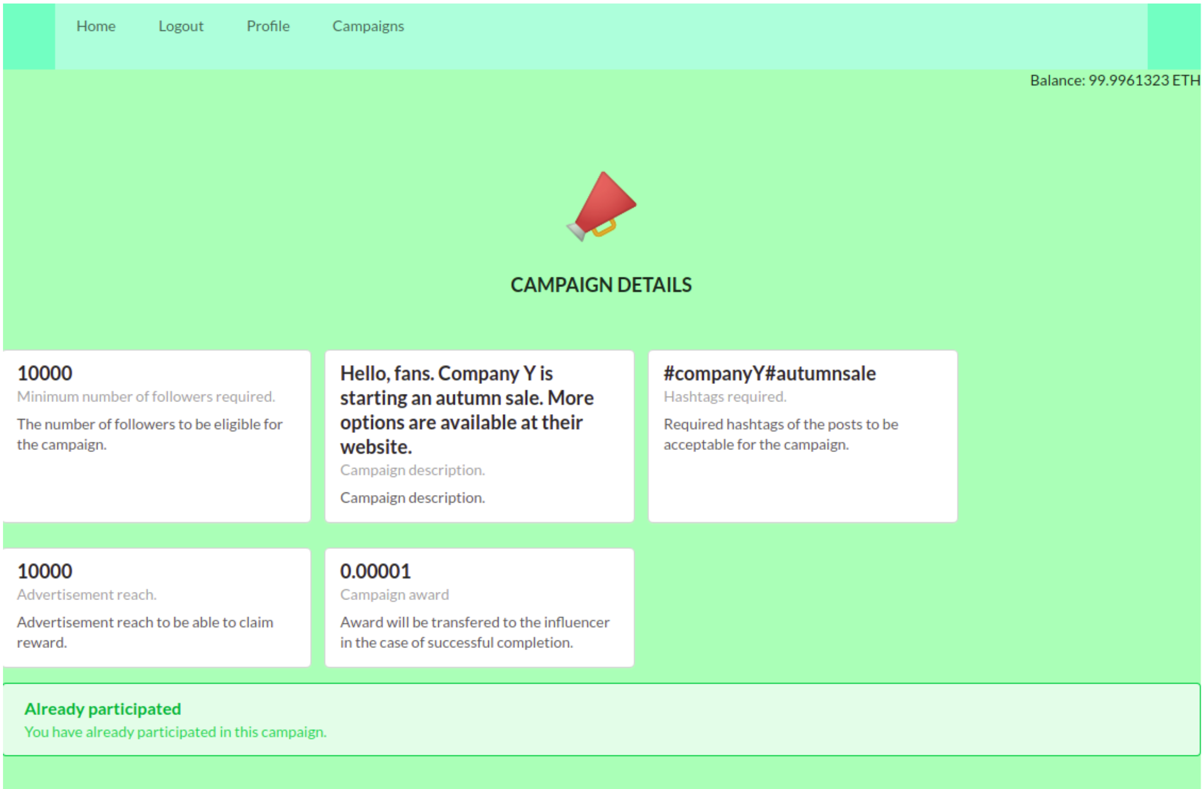

Figure 4.3.1: The information message shows that the macro-influencer or micro-influencer has already participated in the campaign.

That is why we use *require* to check if the message sender exists in the *alreadyAppliedInfluencers* map. If it exists, then it throws a require-type of exception, and the transaction is reverted. However, in the front end application, we also need to check if the influencer account address exists in the map. If it does, then we hide the *apply For Campaign* button. Instead, we show a message that informs influencer regarding previous participation in the campaign (see Figure 4.3.1). That message is displayed only if the influencer is involved in the whole process for that specific campaign, including claiming the reward. Checking the existence of the message sender only on the client-side is not enough for security. If *require* restriction is not used, some other contracts in the blockchain can call the campaign contract, although they should not be authorized to call. Thus with the help of *require*, those kinds of calls are prevented. The requirement to apply for the campaign is to have enough followers. When the brand or macro-influencer creates a campaign, they specify the minimum number of the followers (*minimumNoOfFollowers*) to be eligible for the campaign. If the number of followers of the influencer is less than *minimumNoOfFollowers*, they will not be able to apply for the campaign (see Figure 4.2.7).

4.3.2 Withdraw

The influencers can withdraw from the campaigns. In order to be able to withdraw from the campaign, they should have already applied for the campaign, which means the influencer should exist in the *currentInfluencers* map. In that case, the button will appear to withdraw from the campaign (see Figure 4.3.2). If they withdraw from the campaign, then the maps will be updated, the influencer's account address will be removed from the map of the current influencers and outstanding payments. When they withdraw from the campaign, by removing them from the existing maps, we let the influencers to reapply for the campaign if they decide to.

```
    function withdrawFromCampaign() public payable {
        require(currentInfluencers[msg.sender]);
        currentInfluencers[msg.sender] = false;
        outstandingPayments[msg.sender] = false;
    }

```

4.3.3 Perform Requirements

After applying for the campaign, they exist in the current influencers map, and they can perform the requirements of the campaign in order to be able to claim the reward for the campaign. After they perform requirements, two processes are happening. Firstly, the post is shared on the Instagram simulation, and then

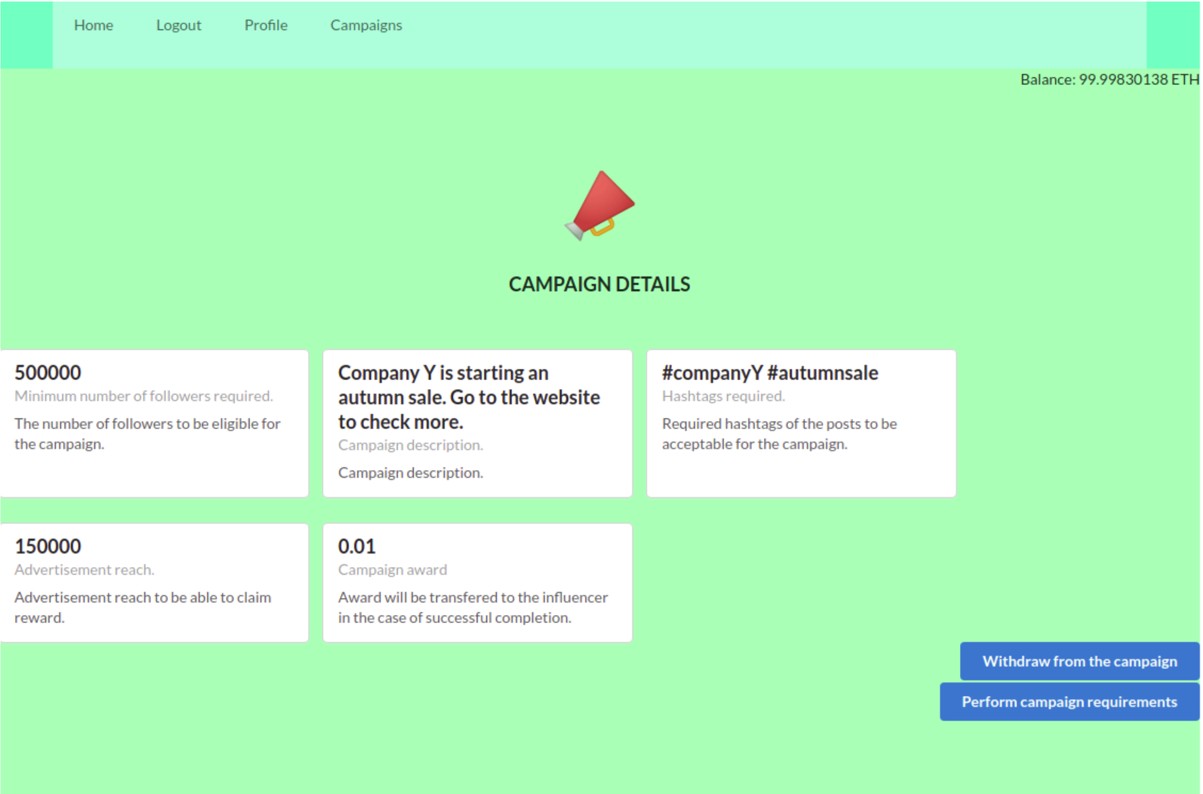

Figure 4.3.2: After applying for any campaign, an influencer can withdraw from that campaign later on.

performedCampaignRequirements method of the smart contract is called. In order for this function to result in a successful call, the influencer address should exist in the *currentInfluencers* map. After checking this condition by *require*, the influencer account address is added to outstanding payments.

```
function performedCampaignRequirements() public {
    require(currentInfluencers[msg.sender]);
    outstandingPayments[msg.sender] = true;
}

```

Performing requirements is behaving differently for macro-influencers and micro-influencers. When a macro-influencer performs requirements of the campaign created by a brand, the post is shared to the Instagram simulation. At the same time, a sub-campaign is created by the macro-influencer (see Figure 4.3.4). This sub-campaign is available for micro-influencers to participate in. On the contrary, when a micro-influencer performs requirements of the sub-campaign created by a macro-influencer, the post is shared to the Instagram simulation, no contract is created. That is why when the micro-influencer performs the requirements of the campaign, they don't need to

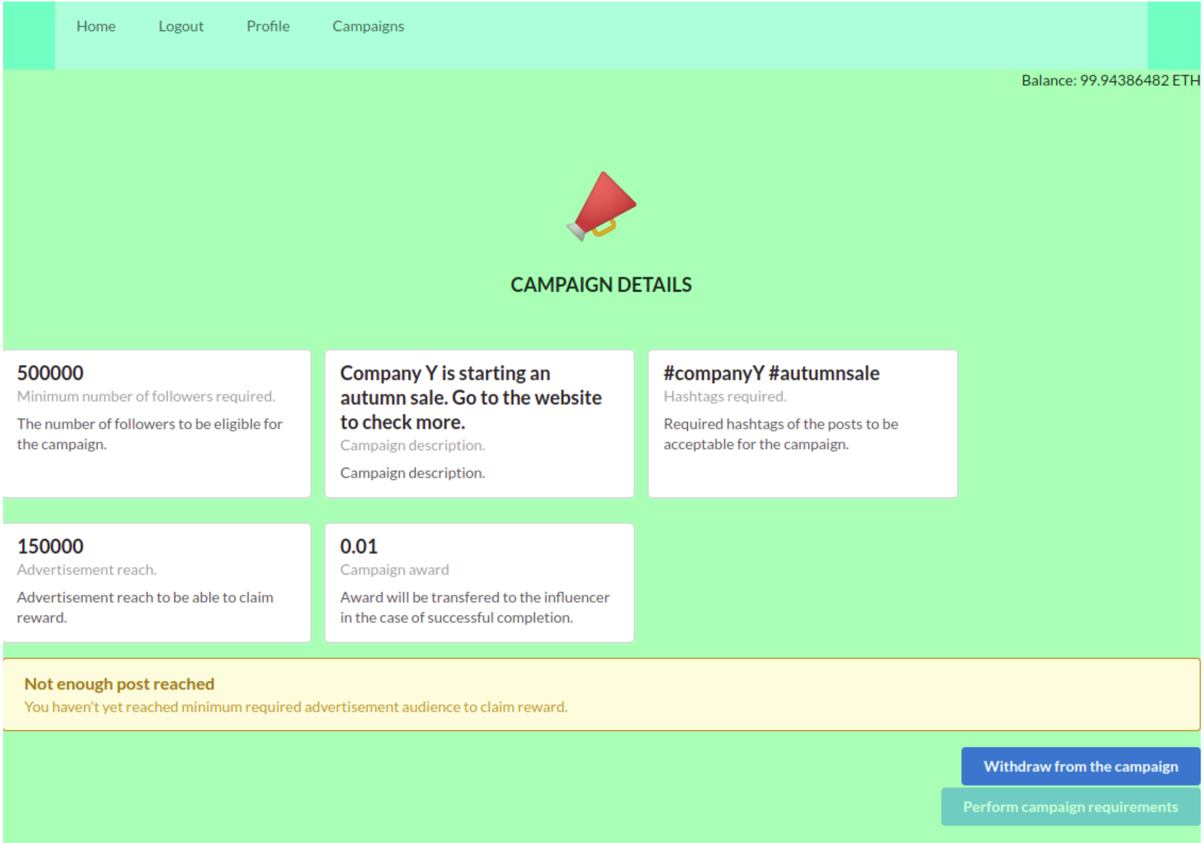

Figure 4.3.3: The message which informs the influencer that the target audience reach is not achieved.

specify campaign details as macro-influencers do (see Figure 4.3.5).

4.3.4 Claim Reward

In order to be able to claim the campaign award, two requirements must be met. First, the influencer account should exist in both *currentInfluencers* and *outstandingPayments* map. By using the *require* function of the Solidity, we achieve that behavior. The second requirement is to reach the required number of audience reach. It means that the post shared on social media must have a minimum audience reach (*advertisementReach*) specified by the campaign owner when the smart contract is created. If they haven't reached the requirement audience target, then they will not be able to claim the reward (see Figure 4.3.3). The audience reach for macro-influencer and micro-influencer is calculated differently. For a macro-influencer, it is considered the sum of the engagement of the posts of a macro-influencer and the engagement of the posts of a micro-influencer who applied for the sub-campaign that macro-influencer have created. On the other hand, since a micro-influencer can't

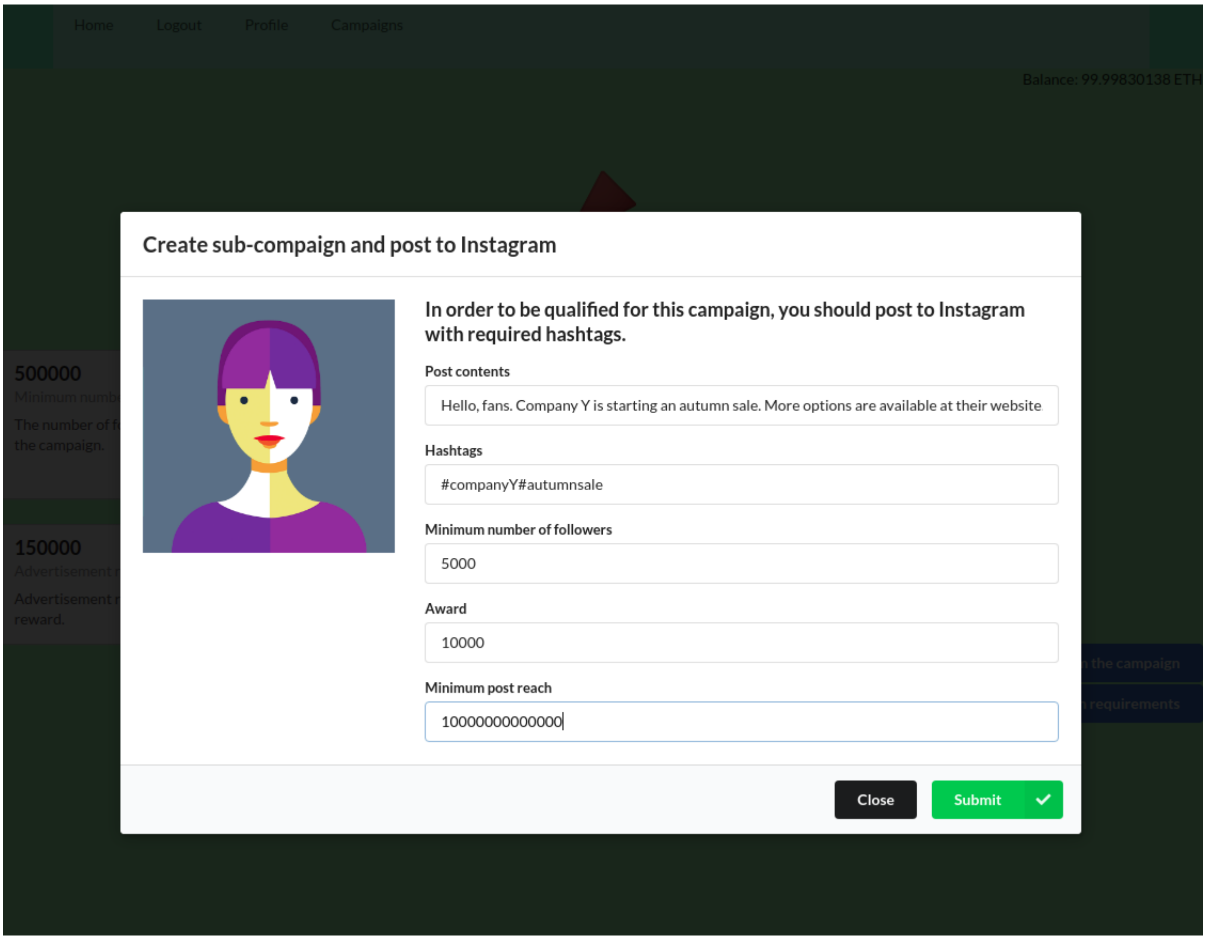

Figure 4.3.4: The page where a macro-influencer creates a sub-campaign and share a post on Instagram.

create the sub-campaign, the audience reach is simply the engagement of the posts they have shared on social media. If these two requirements are met, then the button to claim the reward will be available on the campaign details page (see Figure 4.3.6). After reaching the required target, when the claim reward function (*claimTokens*) is called successfully, the specified reward for the campaign is transferred to the influencer's account.

```
        require(outstandingPayments[msg.sender]);
        require(currentInfluencers[msg.sender]);

```

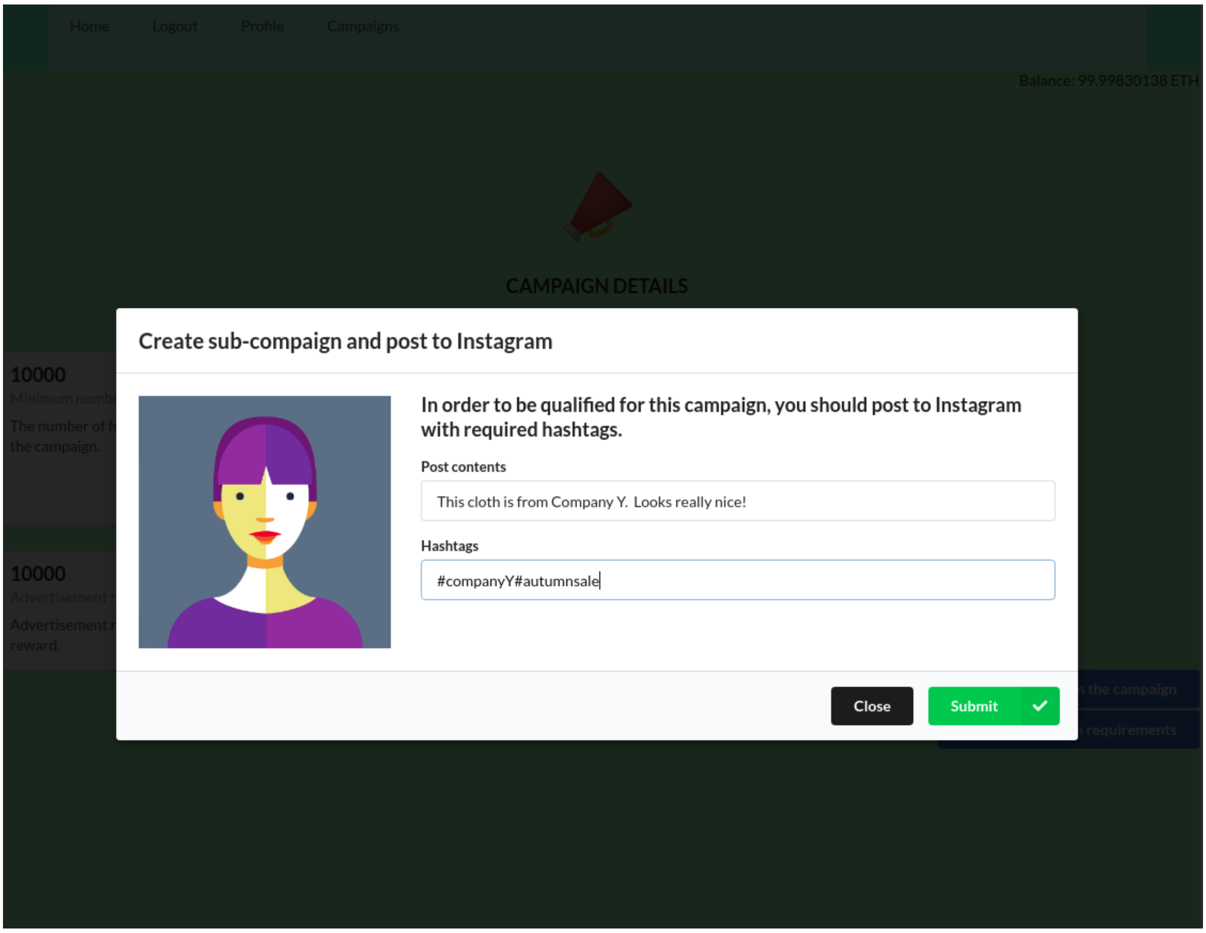

Figure 4.3.5: The page where a micro-influencer share a post on Instagram.

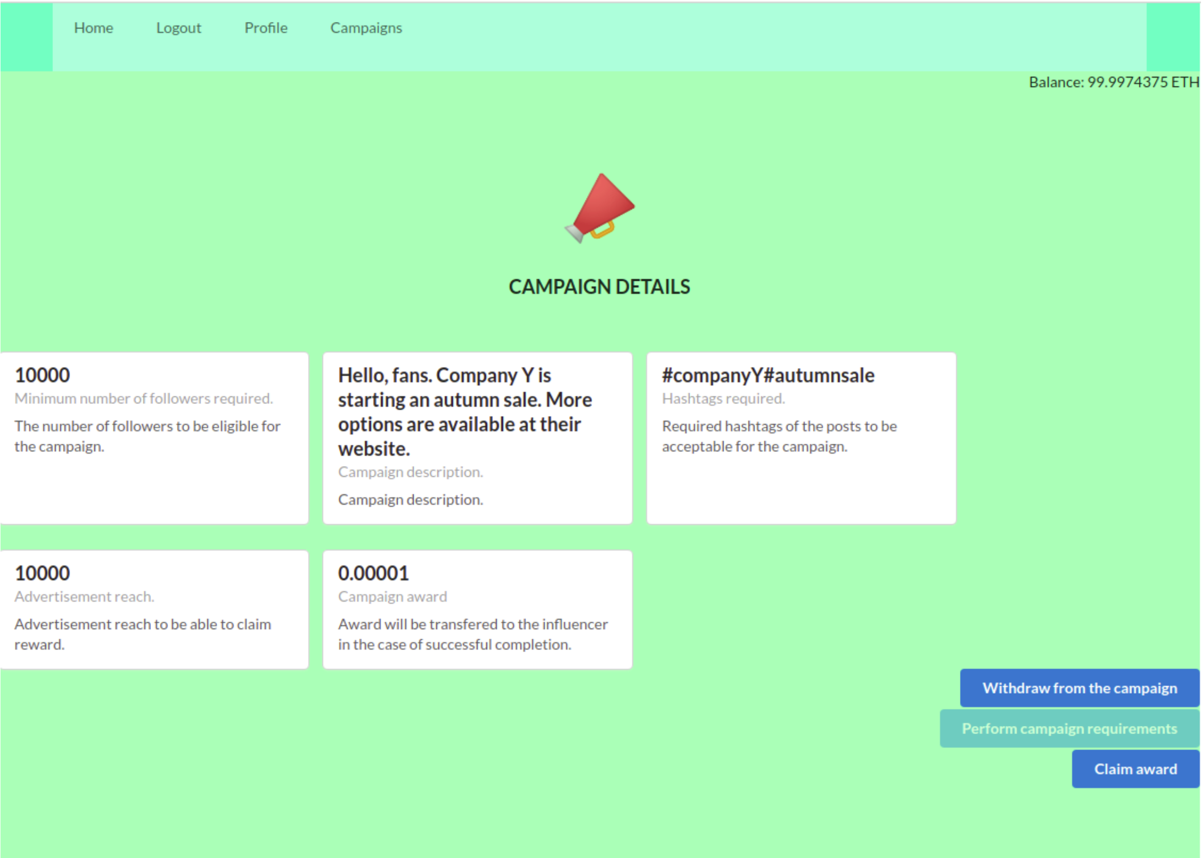

Figure 4.3.6: The page where a macro-influencer or a micro-influencer claim reward.

Chapter 5

Analysis and Results

This chapter describes the experimental results and analysis of the results.

5.1 Experimental results

The proposed smart contract has been deployed on the Ropsten Test Network for evaluation. In order to connect to the Ropsten Test Network, firstly, with the help of Metamask, a few accounts have been created in Ropsten. And in order to deploy the smart contract, Remix is used. The Remix IDE is connected to Metamask with the injected Web3. The main metrics that we wanted to measure were the transaction cost for the smart contracts' actions. That was an important measurement to see whether the smart contract cost is higher than traditional advertisement costs. In order to prevent the financial costs, the tests have been performed on the test network.

The deployment of the contract factory was the first step to be able to create individual contracts. The deployment of the contract factory resulted in the new contract creation for the campaign factory. The total cost of the campaign factory deployment was 0.004016 ETH (see Table 5.1.1). This transaction is to happen only once when the platform is available for production. So this cost is not going to happen in the later stage of the process. Since we need to deploy the campaign factory only once, after the deployment of the factory, we will use that factory contract to create new individual campaigns.

There is an available platform (Etherscan) to use to explore the transactions. Etherscan is mostly used as a search engine. It is possible to search for the transactions and see

Contract Factory Deployment Costs	
Total price	0.004016 ETH
Gas price	2.385331515
Gas limit	1683686
Processing time	31s

Table 5.1.1: The costs of the deployment of the campaign factory contract in Ropsten Test Network

Campaign creation costs	
Total price	0.002847 ETH
Gas price	2.348464876
Gas limit	1212433
Processing time	34s

Table 5.1.2: The costs of the new campaign creation

the transaction details like the status, gas price, gas limit, and total costs.

After creating and deploying the campaign factory, the new campaign creation has been tested to measure the cost. This cost happens every time a brand creates a new campaign. The amount is charged from the brand account in the Ethereum blockchain network. The total cost for the campaign creation is 0.002847 ETH (see Table 5.1.2).

The fee charged from the influencers for the 4 actions below:

- Apply for the campaign
- Withdraw from campaign
- Perform campaign requirements
- Claim reward

The total cost to apply for a campaign is 0.000199 ETH (see Table 5.1.3). This amount is charged once the influencer applies for a campaign. The processing time of the transaction on the Ropsten Test Network is 32 seconds.

Costs of applying for campaign	
Total price	0.000199 ETH
Gas price	2.348464876
Gas limit	84931
Processing time	32s

Table 5.1.3: The costs of applying for the campaign

Costs of withdrawing from a campaign	
Total price	0.000035 ETH
Gas price	0.00000000234
Gas limit	31,395
Processing time	17s

Table 5.1.4: The costs to withdraw from the already applied compaign

Costs of performing campaign requirements	
Total price	0.000101 ETH
Gas price	2.348464876
Gas limit	43194
Processing time	18s

Table 5.1.5: The costs associated with performing campaign requirements

The total cost to withdraw from a campaign is 0.000035 ETH (see Table 5.1.4). The processing time of the transaction on the Ropsten Test Network is 17 seconds.

The total cost to perform campaign requirements is 0.000101 ETH, that is the cheapest action in the whole process (see Table 5.1.5). The transaction processing time is 18 seconds, which is also much less than the processing time for other actions.

To claim the reward for the campaign, 0.000199 ETH is charged from the influencer's balance. (see Table 5.1.6) The transaction processing time was approximately 18 seconds for the action.

5.2 Evaluation

As calculated in the previous section, the cost paid by a brand for a campaign creation is approximately 0.002847 ETH. This amount is much less than the traditional advertisement costs where brands spend much more money on administrative works than this. The total cost of involvement in the campaign application from start to the claiming reward is 0.000499 ETH. The cost to withdraw from a campaign is not

Costs of claiming reward	
Total price	0.000199 ETH
Gas price	2.348464876
Gas limit	64312
Processing time	18s

Table 5.1.6: The costs associated with claiming token

calculated in the total cost of the process since the withdrawal is not a part of the happy scenario in the proposed solution.

$$totalCost = applicationCost + performRequirementsCost + claimRewardCost$$

Comparing the total amount to participate in a campaign with the reward the influencers get in return for promoting products, the cost to participate in the campaign is acceptable. According to the Financial Times [20], influencers with around 100,000 followers on Instagram can charge about £2,000 per post, while celebrity influencers with between 4m and 20m followers can charge around £5,000-£13,000. Compared to these figures, the cost of participating in the campaign is adequate.

In contrast to the existing similar solutions, in the framework proposed, micro-influencers involved in the campaign process. Involvement of micro-influencers requires additional communications with the smart contract since when macro-influencers perform requirements, in order to include micro-influencers in the campaigns, a sub-campaign is created. The experiment results show that, even with the involvement of those communications, considering the time and cost, the proposed framework is viable.

Chapter 6

Conclusion and Future Work

This Chapter discusses the results and what conclusions can be made in relation to the report's problem statement. The limitations of the study and possible future work are also described in this Chapter.

6.1 Conclusion

Based on the result of the testing on Ropsten Network, we can conclude that the proposed solution is feasible for the mentioned problems. The proposed solution is involving the micro-influencers in the campaign process also. By involving in the campaign process, micro-influencers also share a post on social media, which can help solving the problem of increased returns in online shops due to the fitting problem. So the advertised product is not only advertised on the macro-influencers but also the micro-influencers. The whole process is automated by communication with social media simulation with HTTP requests. The transaction costs of the smart contracts' actions are not high numbers, which allows the proposed solution to be applicable for real-world scenarios.

6.2 Limitation

The limitation of this project is the proposed solution hasn't been tested and evaluated in the main Ethereum Network due to the financial costs involved for the actions on the contract deployed. That is why the metrics evaluations have been performed on the Ropsten Test Network.

6.3 Future Work

Future work after this degree project can be to analyze the proposed solution's behavior in the main Ethereum network to see the transaction cost difference between the main network and Ropsten Test Network.

Bibliography

[1] *A Next-Generation Smart Contract and Decentralized Application Platform*. https://github.com/ethereum/wiki/wiki/White-Paper. Accessed: 2020-04-25.

[2] Amaratunga, D, Sarshar, M, and Newton, R. *Quantitative and qualitative research in the built environment: application of "mixed" research approach*. MCB UP Ltd, 2002, pp. 17–31. DOI: https://doi.org/10.1108/00438020210415488.

[3] *AP NEWS*. https://apnews.com/press-release/pr-accesswire/9520cbbdfc20b8f6a28eaf69b10c8a4a. Accessed: 2020-11-11.

[4] *APPICS Official Website*. https://appics.com/. Accessed: 2020-05-16.

[5] Asad Ali Siyal Aisha Zahid Junejo . Muhammad Zawish . Kainat Ahmed . Aiman Khalil, Georgia Soursou. "Applications of Blockchain Technology in Medicine and Healthcare: Challenges and Future Perspectives". In: (Jan. 2019). DOI: https://doi.org/10.3390/cryptography3010003.

[6] Bach, L. M., Mihaljevic, B., and Zagar, M. "Comparative analysis of blockchain consensus algorithms". In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2018, pp. 1545–1550.

[7] *Beigepaper: An Ethereum Technical Specification*. https://cryptorating.eu/whitepapers/Ethereum/beigepaper.pdf. Accessed: 2020-04-27.

[8] Bitcoin.org. *Bitcoin Developer Reference*. URL: https://bitcoin.org/en/developer-reference (visited on 02/18/2020).

[9] *Blockchain in Insurance. Why should you care?* https://www2.deloitte.com/ca/en/pages/financial-services/articles/blockchain-in-insurance.html. Accessed: 2020-10-17.

[10] Chen, Wubing, Xu, Zhiying, Shi, Shuyu, Zhao, Yang, and Zhao, Jun. "A Survey of Blockchain Applications in Different Domains". In: *Proceedings of the 2018 International Conference on Blockchain Technology and Application*. ICBTA 2018. Xi'an, China: Association for Computing Machinery, 2018, pp. 17–21. ISBN: 9781450366465. DOI: 10.1145/3301403.3301407. URL: https://doi.org/10.1145/3301403.3301407.

[11] Dib, Omar, Brousmiche, Kei-Leo, Durand, Antoine, Thea, Eric, and Hamida, Elyes Ben. "Consortium Blockchains: Overview, Applications and Challenges". In: *International Journal on Advances in Telecommunications* 11 (2018), pp. 51–64. URL: http://www.iariajournals.org/telecommunications/tocv11n12.html.

[12] *Digital 2019*. https://wearesocial.com/global-digital-report-2019. Accessed: 2020-05-15.

[13] *Digital Certificates Project*. http://certificates.media.mit.edu/. Accessed: 2020-10-17.

[14] *ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER*. https://files.gitter.im/ethereum/yellowpaper/VIyt/Paper.pdf. Accessed: 2020-04-27.

[15] *ganache*. URL: https://www.trufflesuite.com/docs/ganache/overview (visited on 10/04/2020).

[16] Gopie, Nigel. *What are smart contracts on blockchain?* 2018. URL: https://www.ibm.com/blogs/blockchain/2018/07/what-are-smart-contracts-on-blockchain/ (visited on 02/03/2020).

[17] Guegan, Dominique. *Public Blockchain versus Private blockhain*. Documents de travail du Centre d'Economie de la Sorbonne 2017.20 - ISSN : 1955-611X. Apr. 2017. URL: https://halshs.archives-ouvertes.fr/halshs-01524440.

[18] Gupta, Suyash and Sadoghi, Mohammad. "Blockchain Transaction Processing". In: (May 2018). DOI: 10.1007/978-3-319-63962-8_333-1.

[19] Hennig-Thurau, Thorsten, Gwinner, Kevin P., Walsh, Gianfranco, and Gremler, Dwayne D. "Electronic word-of-mouth via consumer-opinion platforms: What motivates consumers to articulate themselves on the Internet?" In: *Journal of Interactive Marketing* 18.1 (2004), pp. 38–52. ISSN: 1094-9968. DOI: https:

//doi.org/10.1002/dir.10073. URL: http://www.sciencedirect.com/science/article/pii/S1094996804700961.

[20] *How Instagram influencers turn followers into dollars*. https://www.ft.com/content/fc964254-155f-11e7-b0c1-37e417ee6c76. Accessed: 2020-12-17.

[21] *IndaHash Official Website*. https://indahash.com/. Accessed: 2020-05-16.

[22] *IndaHash Whitepaper*. https://s3.eu-central-1.amazonaws.com/idh-files/ico-landing/indahash_whitepaper_ico.pdf. Accessed: 2020-06-16.

[23] Kaulbars-Staudinger, Emilia. “Consumer online shopping behavior affected by influencer marketing - with a focus on sustainability”. In: (2019). URL: http://urn.fi/URN:NBN:fi:amk-2019052611960.

[24] Li Aiya Wei Xianhua, He Zhou. “Robust Proof of Stake: A New Consensus Protocol for Sustainable Blockchain Systems”. In: (Apr. 2020). DOI: https://doi.org/10.3390/su12072824.

[25] Lin, Iuon-Chang and Liao, Tzu-Chun. “A Survey of Blockchain Security Issues and Challenges”. In: *I. J. Network Security* 19 (2017), pp. 653–659.

[26] Luu, Loi, Chu, Duc-Hiep, Olickel, Hrishi, Saxena, Prateek, and Hobor, Aquinas. “Making Smart Contracts Smarter”. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’16. Vienna, Austria: Association for Computing Machinery, 2016, pp. 254–269. ISBN: 9781450341394. DOI: 10.1145/2976749.2978309. URL: https://doi.org/10.1145/2976749.2978309.

[27] McKinnon, A. “Carbon auditing the ”Last Mile”: Modelling the environmental impacts of conventional and online non-food shopping”. English. In: *Proceedings of the 14th Annual Logistics Research Network Conference, Cardiff*. Ed. by A Beresford. Annual Conference of the Logistics Research Network 2009 ; Conference date: 09-11-2009 Through 11-11-2009. 2009.

[28] Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., and Qijun, C. “A review on consensus algorithm of blockchain”. In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Oct. 2017, pp. 2567–2572. DOI: 10.1109/SMC.2017.8123011.

[29] Mohanty, Debajani. "Deploying Smart Contracts". In: *Ethereum for Architects and Developers: With Case Studies and Code Samples in Solidity*. Berkeley, CA: Apress, 2018, pp. 105–138. ISBN: 978-1-4842-4075-5. DOI: 10.1007/978-1-4842-4075-5_4. URL: https://doi.org/10.1007/978-1-4842-4075-5_4.

[30] Nakamoto, Satoshi. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008. URL: https://bitcoin.org/bitcoin.pdf (visited on 02/27/2020).

[31] *PATRON Whitepaper*. https://patron-influencers.com/pdf/Whitepaper_2019_April_Public_compressed.pdf. Accessed: 2020-11-11.

[32] Ranga, Mamta and Sharma, Deepti. "INFLUENCER MARKETING- A MARKETING TOOL IN THE AGE OF SOCIAL MEDIA". In: 2014.

[33] Raval, Siraj. *Decentralized Applications. Harnessing Bitcoin's Blockchain Technology*. O'reilly Media, 2016. ISBN: 9781491924549.

[34] *Remix - Ethereum IDE*. URL: https://remix.ethereum.org/ (visited on 10/03/2020).

[35] *Ropsten Testnet Explorer*. URL: https://ropsten.etherscan.io/ (visited on 12/08/2020).

[36] Savelyev, Alexander. "Contract law 2.0: 'Smart' contracts as the beginning of the end of classic contract law". In: *Information & Communications Technology Law* 26.2 (2017), pp. 116–134. DOI: 10.1080/13600834.2017.1301036. eprint: https://doi.org/10.1080/13600834.2017.1301036. URL: https://doi.org/10.1080/13600834.2017.1301036.

[37] Sharples, Mike and Domingue, John. "The Blockchain and Kudos: A Distributed System for Educational Record, Reputation and Reward". In: *Adaptive and Adaptable Learning*. Ed. by Katrien Verbert, Mike Sharples, and Tomaž Klobučar. Cham: Springer International Publishing, 2016, pp. 490–496. ISBN: 978-3-319-45153-4.

[38] *Solidity*. URL: https://solidity.readthedocs.io/en/v0.7.3/ (visited on 10/03/2020).

[39] Swan, Melanie. *Blockchain: Blueprint For A New Economy*. O'reilly Media, 2015. ISBN: 9781491920497.

[40] Szabo, Nick. "Smart Contracts". In: *Virtual School* (1994).

[41] *The Autho Authentication API*. URL: `https://www.auth0.com/` (visited on 12/07/2020).

[42] Vujičić, D., Jagodić, D., and Ranđić, S. "Blockchain technology, bitcoin, and Ethereum: A brief overview". In: *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*. 2018, pp. 1–6.

[43] *web3.js - Ethereum JavaScript API*. URL: `https://web3js.readthedocs.io/en/v1.2.6/` (visited on 10/03/2020).

[44] *WOMEN ARE THE NEW MEDIA. How influencers became publishers*. `https://labs.indahash.com/wp-content/uploads/2017/06/indaHash_LABS_report_2017.pdf`. Accessed: 2020-05-16.

[45] Yaga, Dylan, Mell, Peter, Roby, Nik, and Scarfone, Karen. "Blockchain technology overview". In: (Oct. 2018). DOI: `10.6028/nist.ir.8202`. URL: `http://dx.doi.org/10.6028/NIST.IR.8202`.

[46] Yang, R., Murray, T., Rimba, P., and Parampalli, U. "Empirically Analyzing Ethereum's Gas Mechanism". In: *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*. 2019, pp. 310–319.

[47] ZHANG, YAKUN. "Product Returns in a Digital Era: The Role of Multidimensional Cognitive Dissonance, Regret, and Buying Context in the Post-purchase Appraisal Process". In: (2018).

[48] Zheng, Z., Xie, S., Dai, H., Chen, X., and Wang, H. "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends". In: *2017 IEEE International Congress on Big Data (BigData Congress)*. 2017, pp. 557–564. DOI: `10.1109/BigDataCongress.2017.85`.

[49] Zheng, Zibin, Xie, Shaoan, Dai, Hong-Ning, Chen, Weili, Chen, Xiangping, Weng, Jian, and Imran, Muhammad. "An overview on smart contracts: Challenges, advances and platforms". In: *Future Generation Computer Systems* 105 (2020), pp. 475–491. ISSN: 0167-739X. DOI: `https://doi.org/10.1016/j.future.2019.12.019`. URL: `http://www.sciencedirect.com/science/article/pii/S0167739X19316280`.

[50] Zheng, Zibin, Xie, Shaoan, Dai, Hong-Ning, Chen, Xiangping, and Wang, Huaimin. "Blockchain challenges and opportunities: A survey". In: *International Journal of Web and Grid Services* 14 (Oct. 2018), p. 352. DOI: `10.1504/IJWGS.2018.095647`.

TRITA-EECS-EX-2021:11