Decentralized Meetup: Origin, Design, and Future

Xianxin Zeng, Xiaodi Duan

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
Faculty of Applied Science & Engineering, University of Toronto

xianxin.zeng@mail.utoronto.ca
xiaodi.duan@mail.utoronto.ca

Abstract: This paper is the final report for CSC2125. In the final project, we implemented a DApp called Meetup. The idea comes from the popular centralized application Meetup. The user can create, join or quit events in this application through the smart contract. There will be a cost for using these features, and users need to buy tokens with ETH in the application. When the participation is successfully confirmed, the user will get rewarded. Tokens can be used for purchasing items or withdraw at a different exchange rate. We completed these functionalities and discuss the future work at the end of this paper.

1. Introduction

1.1. Origin: Centralized Meetup

Meetup is a popular application founded in June 2002. It is a service used to organize online groups that host in-person and virtual events for people with similar interests. There are almost 35 million Meetup users as of 2017. A user can be a member of multiple groups or RSVP for any number of events. Most of the users are using this web application to find friends, share a hobby, and network with others. Unlike other social media sites, Meetup users do not have followers or other direct connections such as texting with each other. [1]

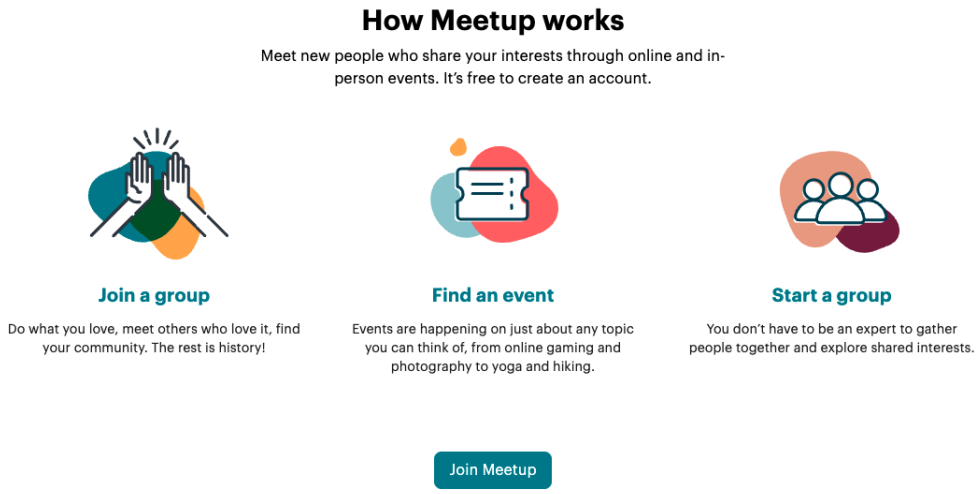

Fig. 1. Meetup Website

From the Meetup website in Fig. 1, we can see how users can start working with Meetup. After signing up, users need to find a group that matches their interests. When they join a group, they can see all the events inside this group. The events are varied, including online events such as gaming or chatting, and in-person events such as yoga or hiking. Every event will have an image that is related to its content. Users can find an event to join and make some good memory. The user can also start a new group of him/herself, and create an event of their own.

1.2. *Decentralized Application*

A decentralized Application(DApp) is an application that can operate autonomously, typically through the use of smart contracts, that runs on a decentralized computing, blockchain system. [2] One of the main difference between traditional applications and decentralized applications is that DApp operates without human intervention and are not owned by any on entity. Once the smart contracts of a DApp is deployed to the blockchain network, it is unchangeable. All the information are stored on the blockchain rather than a centralized data center. All these features make DApp safe and reliable for its users. There is no need to worry about your data being changed or manipulated by someone else.

Based on these advantages of DApp, how can we modify the Meetup application? The Meetup is now centralized, and all the data and programs are stored in a centralized station. The software is owned by one single company, which may cause some risk of privacy leaking. Someone may retrieve your usage history and analyze it to gain more knowledge about your behavior patterns. Most people do not like their privacy to be used by other people. Also, events that contain sensitive content or have a potentially discriminated target audience might be censored. One solution for this problem is turning this application into a DApp.

We are inspired by this idea and implemented a decentralized Meetup in our final project. The project is now published to Github.[1] In this DApp, most of the functionalities of the centralized Meetup are maintained. Users can create, join and quit events as they wish. In order to incentivize people to use this decentralized version of Meetup, some additional features were introduced to make this application profitable for active users. To create, join or quit events, users need to buy some tokens with ETH. There is an entry fee for creating or joining events. If the user decides to quit an event, only part of the fee will be returned. However, if the participation of the user is confirmed, the user will get the tokens back with some bonus. This means that by honestly hosting and joining events, the user can gain more and more tokens. The tokens can be used to buy some real items in the token shop. If the user is not interested in the commodities in the shop, he/she can also withdraw the tokens into ETH and get the money back. The exchange rate will be slightly different because the tokens can be earned during the usage of this application.

In section 2, we will introduce the design and functionalities we implemented. In section 3, we will discuss some future work of this application.

2. Design and Implementation

2.1. *Frameworks and tools*

The following is the frameworks and tools we used in our application:

1. Truffle: Truffle is a popular development framework for Ethereum. It provides smart contract lifecycle management, scriptable deployment and migrations, somple network management and so on. With Truffle, we can easily write and deploy our smart contracts.

2. React: React is a popular front-end JavaScript library that helps build user interfaces. To help us focus on what makes our application unique, we use the React Truffle Box to start our development. This box comes with everything we need to start using smart contract from a React application.

3. MUI: MUI is a popular React UI library. It provides a robust, customizable, and accessible library of foundational and advanced components, enabling you to build your design system and develop React applications faster.

4. Ganache: Ganache is a software that can help you quickly fire up a personal Ethereum blockchain which you can use to run tests, execute commands, and inspect state while controlling how the chain operates.

5. MetaMask: MetaMask is a software cryptocurrency wallet used to interact with the Ethereum blockchain. It allows users to access their Ethereum wallet through a browser extension or mobile app, which can then be used to interact with decentralized applications.

6. IPFS: The InterPlanetary File System(IPFS) is a protocol and peer-to-peer network for storing and sharing data in a distributed file system. IPFS uses content-addressing to uniquely identify each file in a global namespace connecting all computing devices. [3]

[1] https://github.com/LeonhardE/dapp-meetup

2.2. *Application Design*

Fig. 2 shows the basic design and structure of our decentralized application. The DApp creator compiles the smart contracts under the Truffle framework, and the Truffle migrates the smart contracts to the local Ethereum blockchain initialized by Ganache. The DApp interface is implemented with React. The React front-end will connect to the local blockchain via Web3. When the user visits the website with a browser, MetaMask will connect to the blockchain as well, and the user can choose an address to interact with the DApp. When the user uses the functionalities we implemented in our smart contract, he/she will need MetaMask to confirm transactions and pay ETH. If there is an image uploaded, it will be published to the IPFS through the API. Then a Hash value will be returned and the image can be accessible with the concatenated link.

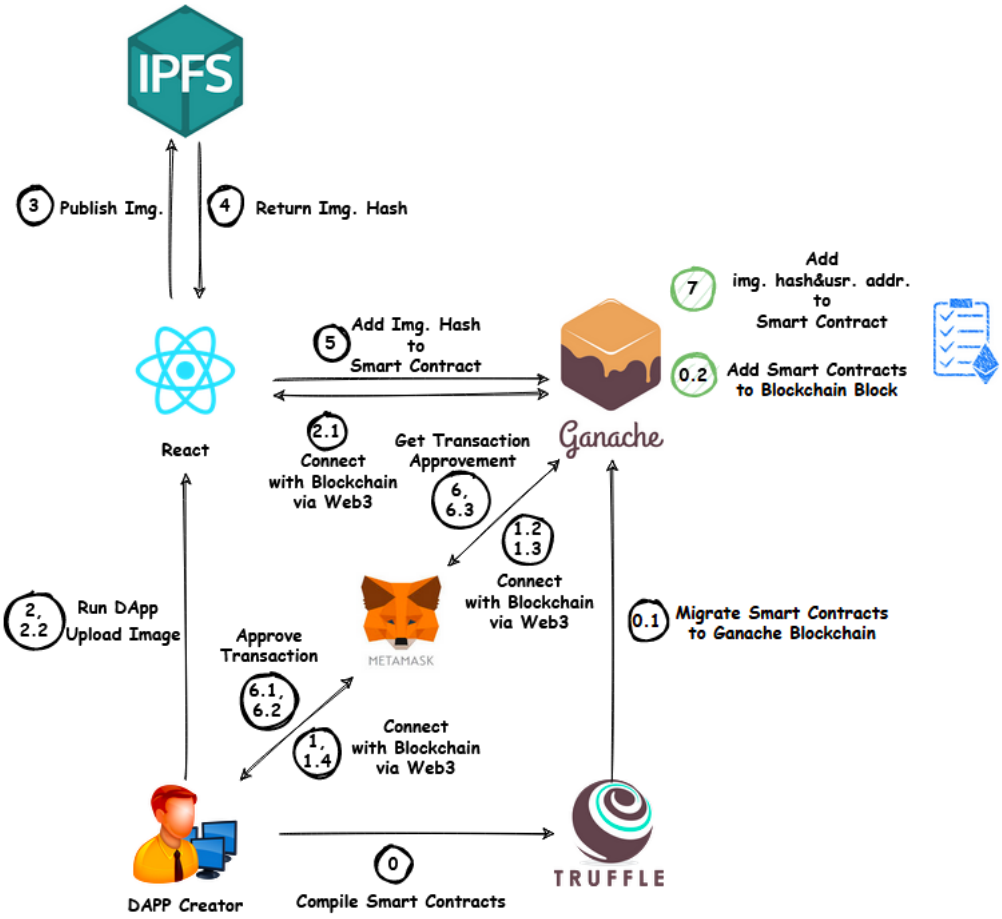

Fig. 2. Decentralized Application Structure [4]

3. Operation and Future Work

3.1. *Application operation*

In this section we talk about how the current application works, and how it generates income for the contract owner.

As was briefly discussed in section 2, users need to exchange ETH to tokens first and it will cost some tokens to create or join events. On confirmation of the event's successful participation/hosting, the tokens will be returned with additional bonus tokens. Tokens can be used later to buy items in the token shop, or can be withdrawn to ETH at a lower exchange rate. With this mechanism, we may create a long-lasting, self-growing, and active community. The contract owner can generate income indirectly when users delete or quit events by returning them fewer tokens than previously charged for. The lower withdrawal exchange rate of token to ETH can also guarantee that we do not lose money.

Fig. 3 summarizes the details of the system's users and their interactions with the system, including six functionalities: create event, join event, quit even, delete event, withdraw token to ETH, and buy items in the token

shop. Both event hosts and successful participants will be rewarded upon each other's confirmation.

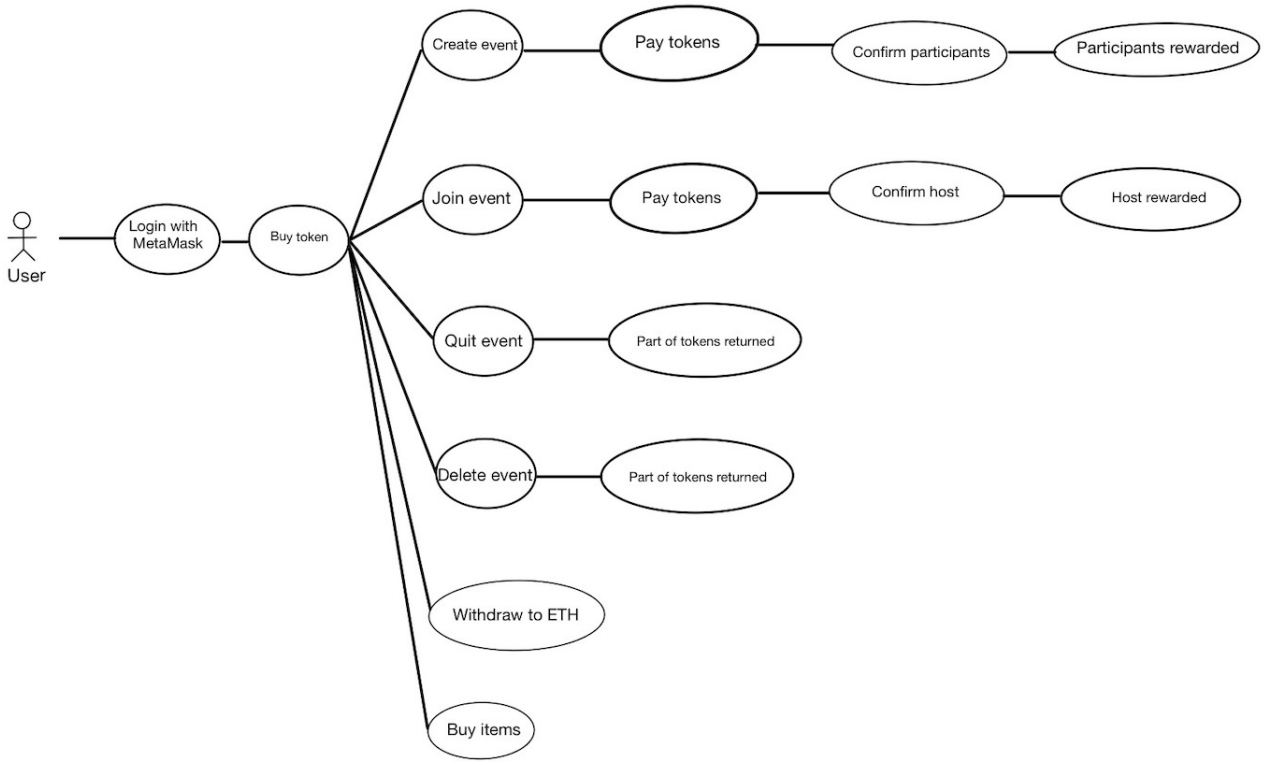

Fig. 3. Use case diagram

3.2. Problems

The current implementation consists of three major problems, which can be fixed in future work.

1. Contract owner is losing money

   The incentive of giving bonus tokens is basically giving out money. Although this seems like a good incentive, the more users there are, the more money the contract owner loses.

2. Trust needed

   The withdraw request must be confirmed and transfer will be launched by the contract owner. The contract owner's wallet serves as a trusted third party here, and the trustless feature is damaged. This cannot be fixed as long as the token component exists.

3. Fraud happens

   In our current implementation, there is a two-way confirmation: after the event, the host can confirm the successful participants, and the participants can confirm that the event was successfully held. However, because we use addresses as accounts, it is likely that dishonest users will use multiple addresses and create fake events which can be confirmed with fake participants.

3.3. Possible solutions

There are some possible solutions to the previous problems.

Following along the popular user incentive systems in the market, we can modify the bonus value to a decreasing value according to users' number of active events so that the longer the user stays in the network, the lower bonus it will be. By doing so, the investment in the start-up phase is acceptable.

To solve the fake-user attack, we may introduce a decentralized identity verification into the program to guarantee that one real person only has one account, or that the accounts belonged to the same identity cannot join its own events.

However, the program is still vulnerable to a fake-event attack (a group of dishonest users confirming their own fake events). The essence of the attack lies in the bonus mechanism: on confirming the event without any centralized way of verifying, all the members in that event will be rewarded, and consequently, the contract owner loses money. Only if the bonus is not generated directly from the contract owner's pocket can we prevent this attack.

Therefore, we need to forfeit the original idea of incentivizing users with bonus, and replace it with punishing users for not completing the event. Tokens still need to be spent in order to create/join events. If most of the participants that the host has confirmed also confirmed the host for holding the real event, the event is regarded honest and successful so that everyone can get their tokens back. On the contrary, if there is a fake event and most confirmed participants say the event is fake, then the host will be punished by deducting his/her/their tokens and the honest participants' entry fees will be returned. The participants who confirmed that the event was fake would be rewarded with some bonus tokens for reporting the dishonest behavior. The sum of bonuses should be equal to the penalty of the host so that we don't lose money.

4. Future work

In order to encourage more users to this application, we can add a decreasing bonus value for each successfully hosted event based on each decentralized-verified identity specified in section 3.2. The bonus could also be proportional to the number of participants as another incentive.

The solutions described in section 3.2 including penalty for fake events and the rewards for reporting can also be implemented to solve the fake-event problem and the dishonesty problem. There could also be APIs designed for human intervention to temporarily solve some potential problems.

References

1. Wikipedia for Meetup: https://en.wikipedia.org/wiki/Meetup.
2. Wikipedia for DApp: https://en.wikipedia.org/wiki/Decentralized_application.
3. Wikipedia for IPFS: https://en.wikipedia.org/wiki/InterPlanetary_File_System.
4. Dapp University: https://www.dappuniversity.com/.