Introduction to Blockchain Technology: Meme Economy Blockchain

Rahul Agrawal, Akshay Katyal, Mehmed Mustafa
Anant Sujatanagarjuna, Steffen Tunkel, Chris Warin

August 2, 2020

Abstract

The meme culture is spread widely over online communities. This project aims to create a novel decentralized platform for the sharing of memes. The *Meme Economy Blockchain* is built up like a virtual marketplace, based on the principles of market economy. Through this, users are rewarded for creating new and innovative memes and also for their intuition about the potential popularity of memes. The current state of the project contains all core functionalities, which are required for the application.

Contents

1 Introduction

A meme is *"an amusing or interesting item (such as a captioned picture or video) or genre of items that is spread widely online especially through social media"* according to Merriam-Webster dictionary [4].

An example for such an item is shown in figure 1 a. This particular joke about procrastination is combined with the pictures of the musician Drake to highlight the message. These pictures, as shown in figure 1 b, are a genre of memes. These are templates which are used over and over again to create new memes. In this work, templates for memes are named 'meme formats', while 'meme' refers to the captioned pictures as in figure 1 a.
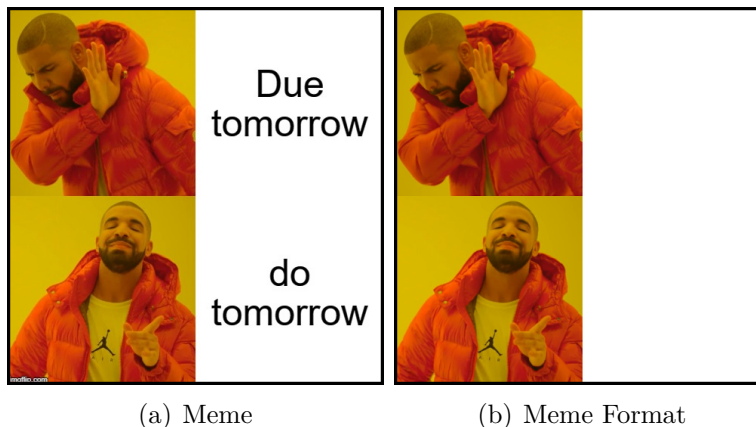

(a) Meme   (b) Meme Format

Figure 1: 'Drake Hotline Bling' as an example for a meme and meme format [3].

Memes are not only spread over classic social media platforms, like Facebook or Twitter, but also on more dedicated platforms like 9GAG or Reddit. One of Reddit's sub-forums is called MemeEconomy [6]. It is a virtual market place for memes, which is implemented with bots on the Reddit server. It served as inspiration for us to create a more free, decentralized version of an economy system for memes, which can therefore be run by the community itself.

The application, which is implemented on a blockchain technology, can be thought of as a virtual stock market for memes. That means, users are able to buy and sell shares of meme formats and speculate on their value. The value is defined by the estimated lifetime of the meme format and its circulation. This is based on the concept that the shareholders of a format get a reward every time it is used for a meme. The basic principles of financial markets are also applied in the approach. Users are motivated to buy a share when its value is low and has a higher potential to increase, and vice versa, to sell when its value is high and expected to go down in the future.

Moreover, users have the chance to invest in particular memes with their upvotes. The revenue a user can gain with an upvote is proportional to the amount of following upvotes by other users. Therefore, a good intuition and early support of a meme is rewarded.

The credits used in the economy system can be gained through smart investments, the creation of new and potentially successful content or the support of the blockchain with the mining of new blocks. The credits do not have any money value. Therefore, they can be interpreted as the user's score.

The motivation of the project is to offer users a fresh and interesting platform for meme sharing by enhancing the concept of the MemeEconomy subreddit. Additionally, the platform encourages the creation of new memes and even new meme genres. By recognizing memes as a form of art, this factor gets even more important. Connected to this, it is also possible to track down the origin of popular meme formats over the blockchain, which means the credit of the work goes where it belongs.

This report aims to give a rather technical introduction to the approach. Section 2 presents the different technologies used for the implementation. Afterwards, section 3 gives an introduction to the MemeEconomy from Reddit as a related work. The main part is the technical description of our approach, outlining the different functional parts of the blockchain, in section 4. Finally, section 5 gives an outlook on further possibilities with the approach before section 6 concludes the report.

2 Foundations

This section gives information regarding the software and interfaces used during the development of our Blockchain application.

2.1 RESTful Applications

A RESTful application is an *Application Programming Interface* (API) that uses HTTP requests to access the resources [1]. An API for an application is code that allows two software programs to communicate with each other.

A RESTful application can also be described as a web service based on REST, abbreviated as **Re**presentational **S**tate **T**ransfer. The application breaks down user requests into a small series of modules, where each module addresses the part of the request. The RESTful application takes JSON strings as the input given as: *{"name": "example"}*. As discussed earlier, a RESTful application uses HTTP methodologies to access the resources (object, file, or block) given as follows:

- GET: Request to retrieve a resource

- PUT: Request to update a resource

- POST: Request to create a new resource

- DELETE: Request to remove a resource

2.2 Flask Framework

Flask is a Python-based web-development framework, also called as *Web Server Gateway Interface* (WSGI) [2]. It was first used as a simple wrapper around Werkzeug and Jinja and has now become the most popular Python web application framework. The Flask framework is used to start a local development server on the user system; this is done by importing **flask run** in the application by the use of the FLASK_APP environment variable.

Flask uses the *route()* decorator to determine which function will execute in the program code. Flask handles the requests and responses between the local server nodes in our blokchain application.

Sample Flask example:

```
1  from flask import Flask
2  app = Flask(__name__)
3  @app.route('/example/')
4  def example():
5      return {'hello': 'world'}
6
7  python ./example.py
8   * Running on http://127.0.0.1:5000/
```

2.3 Postman

Postman is a platform for API development [5]. The Postman interface has a variety of features that help in the building of an API, such as:

- API Client: Helps sending requests within Postman

- Automated Testing: Helps testing the API

- Monitor: Helps to check the performance of the API

- Documentation: Helps to create documentation for our API

The API client allows the user to send requests and view response, generated by the execution of the REST queries within Postman. It also allows the user visualizing data by using the 'visualizer' feature — in our application, Postman helps visualizing the memes and the meme formats. The platform also allows the user to create a test suite for their application, using the 'collections' feature, where each collection consists of multiple HTTP requests.

3 Related Work

In the subreddit */r/memeEconomy*, memes are portrayed as a potential investment where the members can buy, sell, share, and invest in them, based on their potential of becoming a popular meme on the website. In other words, memes and economics are combined in perfect proportion. The official currency of the subreddit is *Meme Coin* (MC).

When a user creates an account, they are given 10,000 MC, as well as a debt of 10,000 MC that they need to pay back. The return on the meme investment is determined by the demand of that meme, and the incentive is calculated in the following way: the higher the meme charts on the subreddit, the more MCs the user earns. The Meme Economy limits the amount of MCs in the economy. It follows a risk-return trade-off pattern which is similar to crypto-currencies.

Members who create a post on /r/MemeEconomy receive MCs in the form of royalties, totalling 5% of all investments on the post. They are paid once in 24 hours after the post

is released, and investments after this 24 hour period will not count towards the 5% bonus. This is a way of rewarding the creators for their work.

4 Approach

4.1 Block Design

Each block inside the chain has the following fields:

- index - the index of the block.

- minerID - the id of the miner. This identification is used when distributing mining rewards.

- previous_hash - the hash of the previous block inside the chain

- proof_of_work - a nonce value (magic number) which makes the hash of the block to match the difficulty pattern

- timestamp - an instance of time showing the creation time of the block

- transaction_counter - the number of transactions inside the block

- transactions - a list which holds all transactions inside the block

The genesis block has a value of "0" on all fields and an empty list for the transactions. The hash value of the genesis block does not match any hash patterns, and when validating the chain, the genesis block is skipped. It should be noted that even if the block is not skipped, since all values inside the genesis block are predefined, the hash value will always be the same value and could be hard coded.

All consecutive blocks, after the genesis block, have their proper field values. The *minerID* field is a special field because it's value is different for each node. Each node assigns its ID to the *minerID* field and tries to find a nonce value with this assigned specific *minerID*. Also each time a new transaction is added to the memory pool of transactions, the finding nonce value process is restarted, since the transactions list is updated.

4.2 Peer-to-Peer Network

In a blockchain peer-to-peer connection, any node connects to any other node in the network collection of nodes. The nodes in the network are connected as shown in figure 2, where each node is connected to all of its neighboring nodes, and each node contains a copy of the complete blockchain. When a new node connects to the blockchain network, all the nodes in the network are notified about the newly connected node in the network. Moreover, the newly connected node receives a copy of the complete blockchain, as well as the list of pending transactions in the blockchain.

In our meme economy application, the peer to peer concept is demonstrated by the use of Postman with the following steps:

- The initial node must run on port 5000 in order to initialize the network. This port is selected specifically to prevent the creation of multiple networks at the same time.

- Other nodes are added on different port addresses.

- A POST request is sent, using the **connect_to_node** method. If the request is successful, a response showing "Connection Successful" on the console is returned with status code 200.

As the connection is successful, the newly connected node receives a copy of the blockchain as well as the pending transactions in the blockchain network.
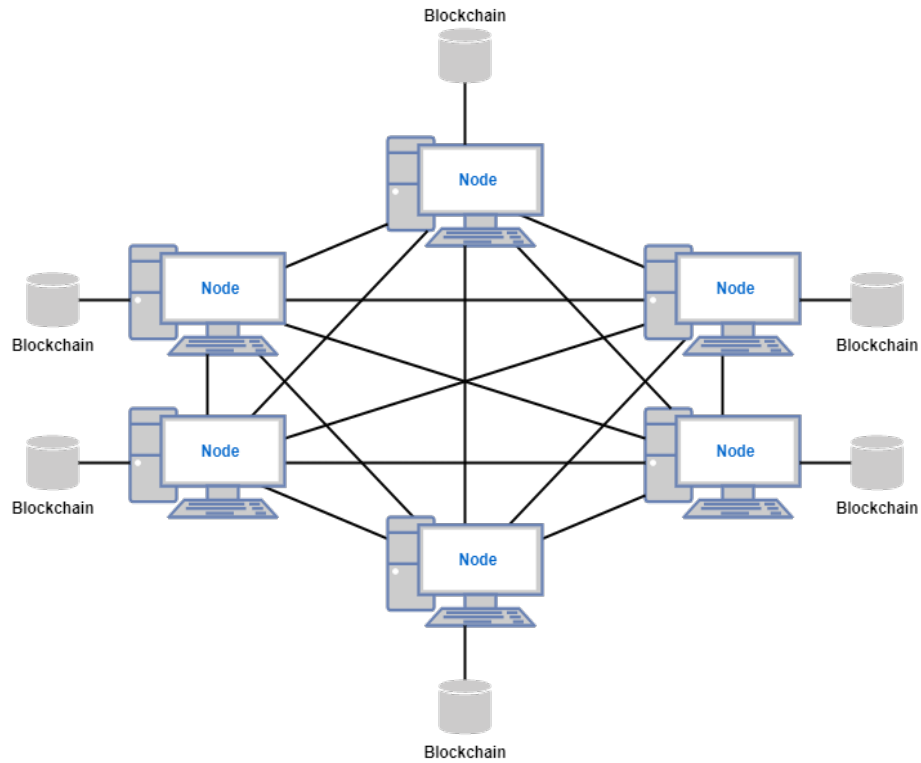

Figure 2: Peer-to-Peer network

4.3   Transactions

As discussed in section 4.1, each block has a list of transactions. There are 5 different kind of possible transactions:

- add_memeFormat - Used to add a new meme format to the network

- add_meme - Used to add a new meme to the network

- add_upvote - Used to upvote a meme inside the network

- sell_ownership - Used to sell the ownership of a meme format

- purchase_ownership - Used to purchase the ownership of a meme format

5

4.3.1 Raw-JSON body input formats of transactions

- add_memeFormat - {"imagePath" : "imagePathValue",
  "name" : "nameValue"}

- add_meme - {"imagePath" : "imagePathValue",
  "name" : "nameValue",
  "memeFormat" : "memeFormatID"}

- add_upvote - {"imageVoteId":"memeID",
  "upvoteID" : "upvoteID"}

- sell_ownership - {"ownershipSaleOfferID" : "ownershipSaleOfferID",
  "memeFormat" : "memeFormatID",
  "saleAmount" : "saleAmount"}

- purchase_ownership - {"ownershipPurchaseID" : "ownershipPurchaseID",
  "ownershipSaleOfferID" : "ownershipSaleOfferID"}

4.3.2 Transaction requirements and validation

This subsection discusses the requirements of each transaction. Once all of the requirements for a specific transaction are met, the transaction is validated and ready to be mined in the next block.

- add_memeFormat - There are not any requirements for this transaction.

- add_meme - Each meme to be uploaded must belong to a specific meme format. This meme format must be already mined inside a block and appended to the chain.

- add_upvote - The meme to be upvoted must be already mined inside a block and appended to the chain. The node which upvotes must have enough credits available in its wallet.

- sell_ownership - The meme format to be sold must be already mined inside a block and appended to the chain. The node which creates this transaction must have the ownership of the meme format to be sold.

- purchase_ownership - The meme format to be purchased must be already offered for a sale, mined and appended to the chain. The purchasing node must have enough credits in order to buy the meme format. The worth of the meme format is specified inside the sell_ownership transaction.

4.4 Mining and Consensus

The consensus mechanism between the nodes is a proof of work based approach. In order to mine a new block the mining node has to solve a search puzzle. For this a json representation of the whole block is created and used as an input for the cryptographic hash function SHA-256. The output of the function has to lay in the target space to successfully mine a block.

The size of the target space and therefore also the difficulty to find a solution is defined by the *difficultyPattern* parameter of the blockchain. By default it is set to $'0000'$ which leads to a success rate of $1/16^4 \approx 0.0015\%$. A correct solution can be found by varying the *proof_of_work* parameter of the block.

The mining of new blocks works in a parallel manner for the nodes. As long as there are transactions to mine, the mining nodes are constantly trying to find a solution for the search puzzle of their blocks in a separate thread. When a node finds a solution for its search puzzle and successfully mine a block, it informs all other nodes about it. In this case the other nodes check the validity of the new block, stop their current mining process and restart based on the new current block chain. When a node adds a new transaction to the network the procedure is similar, starting with the node notifying all other nodes in the network. The other nodes check if the transaction is valid, add it to their transaction list and restart the mining if they can include the transaction to the block they are currently mining.

4.5   Wallet and Rewards

This subsection discusses the role that wallets and rewards play in the operation of the Meme Economy Blockchain.

Every node has a unique and implicit wallet associated with it. Once a node mines a block or issues a transaction that is added to a block, all nodes start to internally and independently track the state of the wallet associated with the node. Every node starts out with an initial wallet amount of 5 credits.

There are two situations in which a node is rewarded credits: either from another node, or from the system. Following are these situations:

- Upvote - When a meme is upvoted, a credit is deducted from the node that issued the upvote. This credit is then distributed in portions to:

  - the Meme Format owner - 30%
  - the Meme Poster - 60%
  - the Meme miner - 10%

  In addition to this, 10% of the upvote credits (which amounts to 0.1 credits) are created independently and rewarded to:

  - the Meme Format miner
  - the Upvote miner
  - every previous Upvoter from previous blocks who upvoted the same Meme

- Ownership Purchase - When a node issues a transaction to purchase ownership, 10% of the sale amount is taken in excess to the sale amount from this node, and is shared evenly between:

  - Sell transaction miner
  - Purchase transaction miner

In effect, the Sell transaction miner and Purchase transaction miner each recieve 5% of the sell amount.

It must be mentioned that all rewards are implicitly credited to the respective node's wallet.

The reason for rewarding upvoters is to incentivize upvoters who find a meme interesting or funny. Over time (as more blocks are mined), an upvoter is rewarded proportionally to how much they contributed to the 'virality' of the meme.

There is no mechanism for a node to cancel a sell transaction. The node is free to issue a purchase transaction for their own sell transaction, if they see fit. In this scenario, they would lose 10% of the amount that they posted the ownership for sale for.

As mentioned earlier, every node is given 5 credits. This is a necessity, since if not, no nodes have any credits to upvote, nor buy ownership.

4.6 Visualization

Every meme and meme format can be visualized by every node once its transaction has been mined. This allows a node to decide whether to upvote the meme, to create their own meme of the given meme format, or, if the meme format is on sale, to buy it.

The blockchain contains the list of all the memes and meme formats that have been mined, and their image is stored as a *base64* encoded string. Since HTML supports the visualization of *base64* encoded images, an HTML tag embedding the encoded string can be returned by the application in such way: `<img src='data:image/jpg; base64, [encoded image string]'>`

5 Outlook

The Meme Economy blockchain project in its surrent state is a good basis for enhancing the eponymous concept. Still, there are a number of improvements that could provide an overall better experience.

First, the support of mediums other than images can be considered. For example, adding *Gif* support would provide a more various experience and more potential for users, while keeping the size of transactions low.

Regarding user experience, adding a *Graphical User Interface* (GUI) would significantly improve the accessibility and usability of the application. Moreover, further developing the application around the prevention of errors would also improve the general experience. For example, collisions can happen for memes or meme formats that share the same name. This does not generate an error that is intelligible for the end-user who might get frustrated.

Regarding the network transactions, it could be interesting to switch to a binary protocol. As of now, images are encoded in *base64*, which does not result in a fixed-size transaction. This means that the required time for mining transactions can be relatively different, depending on whether there is an image in the transaction, and if it is the case, the size of the image. Using a binary protocol would enable a transaction size limit.

Lastly, the application could be made more secure by adding digital signing, e.g. asymmetric key encryption.

6 Conclusion

This report presented the Meme Economy blockchain project. The concept taken from the */r/MemeEconomy* subreddit has been discussed in section 3 and implemented in Python under the Flask framework (see section 2).

Any node connects to any other node in the network in Peer to Peer, and can add transactions (i.e. add a new meme / meme format, upvote a meme, sell of buy a meme format). Transactions of memes or meme formats contain an image that is encoded in base64 and stored as a string. Miner nodes try to mine blocks as soon as there are pending transactions. Nodes can visualize memes (and further decide to upvote or to create their own meme of given meme format). Lastly, nodes have a personal wallet that allow them to make the different transactions, and they are rewarded credits for various situations (e.g. mining a meme format or upvote transaction). These features enable the users to trade in different ways, thus successfully adapting the Meme Economy concept to a blockchain application.

The source code will be released on a public repository under the GNU Affero General Public License. In the future, it can be envisaged to have open collaboration to further improve the concept and invite the internet community to use it as a decentralized platform to create memes.

Abbreviations and Acronyms

API *Application Programming Interface*

WSGI *Web Server Gateway Interface*

MC *Meme Coin*

GUI *Graphical User Interface*

List of Tables

List of Figures

References

[1] RESTful API. What is REST, 2020. Available online: `https://restfulapi.net/`, last retrieved 25/07/2020.

[2] Flask. Documentation, 2020. Available online: `https://flask.palletsprojects.com/en/1.1.x/`, last retrieved 25/07/2020.

[3] Imgflip. Meme templates, 2020. Available online: `https://www.imgflip.com/memetemplates`, last retrieved: 16/07/2020.

[4] Merriam-Webster. Meme, 2020. Available online: `https://www.merriam-webster.com/dictionary/meme`, last retrieved: 23/07/2020.

[5] Postman, 2020. Available online: `https://www.postman.com/`, last retrieved 25/07/2020.

[6] Reddit. Meme economy, 2020. Available online: `https://www.reddit.com/r/MemeEconomy`, last retrieved: 23/07/2020.