Technical White Paper for Dimension Chain

Dr. Monkey D. Luffy

May 15, 2018

1 Introduction

We use blockchain technology [1], hailed as offering immutable proof chains of digital data, to digitize and quantify online and offline activity of fans in support of anime, comic, and idol figures. We propose a blockchain network, Dimension (DMS), to store all relevant activities and make them easily and widely accessible, taking advantage of the essential trustworthiness of blockchain records. DMS will add a layer of abstraction to social network platforms to extract specific user activity. Dimension (DMS) consists of the following components:

- DMS Blockchain is a flexible, single-layer blockchain platform, with an extra shared history database layer, capable of processing a high volume of simultaneous transactions as well as of detecting spam activity and imposing countermeasures to discourage and eliminate spam. Similar to Ethereum, an infinite number of many different smart contract accounts can be issued and run on top of this platform. Smart contract accounts can communicate with each other to support cross-domain transactions.
- DMS Network is a peer-to-peer (P2P) network[2], providing the interface to access DMS blocks and process any validation or transaction request. In this network, any node can validate the authenticity of any other node in the chain.
- DMS Database is a distributed database system that acts as the extra shared his-tory database layer to store previous activity on the social network as an anti-spam mechanism.
- DMS Wallet, a token management platform, enabling micropayments powered by DMS Network and other value transfer requests.

The remainder of this paper explains how these components function and interrelate to provide real world services. It is organized as follows: Section provides an overview of DMS; Section 3 describes the DMS Blockchain in more detail; Section 4 explains DMS Network; Section 5 describes the DMS Shared History Database; Section 6 discusses DMS Wallet; and Section 7 offers our conclusions.

2 Overview of Dimension

Our goal is to record all online and offline activities associated with any anime, comic, or idol figure or creator. This will allow easy tracking of efforts individuals make to support a work or creator. We now limit our focus to online activities on existing social network platforms, including Twitter and Pawoo. Likes, comments, reposts, mentions and any other related actions will all be stored on DMS chain. This will give all the social network users unlimited access to individual fan contributions related to any particular idol or avatar.

Offline activities will be recorded manually on chain by the event organizers. Each offline activity is associated with an online account or identity and it is essential if all these activities could also be recorded and awarded. We will not discuss such activities in detail in this technical white paper since the high cost to fake in such events becomes a natural barrier. The inevitable few cases of faking will not impact the whole system materially.

Online activities include existing social network platforms, e.g. Twitter, Pawoo, etc. Any related activities, including like, comment, repost and mention, will be recorded for future reference.

Both offline and online activities will be stored on DMSchain and thus everyone can access such records without any barrier. We can easily calculate the contribution of each fan to his supporting idol or avatar.

The potential for spam to produce fake idol support[3, 4] could occur when a spammer repeatedly clicks and cancels "like," makes the same or similar comment more than once, reposts, or mentions things multiple times. We will only provide an anti-spam mechanism against the first three activities in the DMS 1.0 edition because we can easily and directly track all likes, comments, and reposts on various social media platforms. By contrast, we are unable to adequately track all mentions unless and

until we can include the entire text of posts in the DMS shared history database used to identify spam. We will seek to support the "mention" feature when we are able to access the entire database of one or more social network platforms.

Another issue presented by online social networks is the high volume of simultaneous activities [5]. Thousands of user requests are made each second and we need to track and store all of them without any loss or problems. Thus, our design must be capable of processing a high volume of simultaneous requests. If the design cannot accommodate unusually high volumes in unusual circumstances, it must be able to stack all requests and set them to automatically process.

We have designed DMS to solve the two issues of counteracting spam and accommodating a high volume of simultaneous requests through a low-level architecture requiring minimum additional computational resources. DMS will also support payment within the blockchain system so no additional mechanism is needed to accomplish value transfers.

3 DMS Blockchain

We start with our core component of the project, the DMS Blockchain. We will describe the detail of block and blockchain architecture, blockchain states, the mechanism to create and validate new blocks, anti-spam mechanism, token rewarding mechanism and poison mechanism which automatically expires unclaimed tokens.

3.1 DMS Blockchain architectures

DMS blockchain is designed as a single-layer blockchain system to store all online and offline activities. For each block creation request, one block is initiated, validated, and then connected to the chain tail. An alternative double-layer blockchain system is also contemplated, which has some advantages over the single-layer design. A virtual machine implementation is also needed since, like Ethereum, we need to support the running of an infinite number of many smart contracts. That implementation is not within the scope of this paper, although we do believe that such a virtual machine could inherit the lightness of weight of EVM [6] and support Python or other major programming languages.

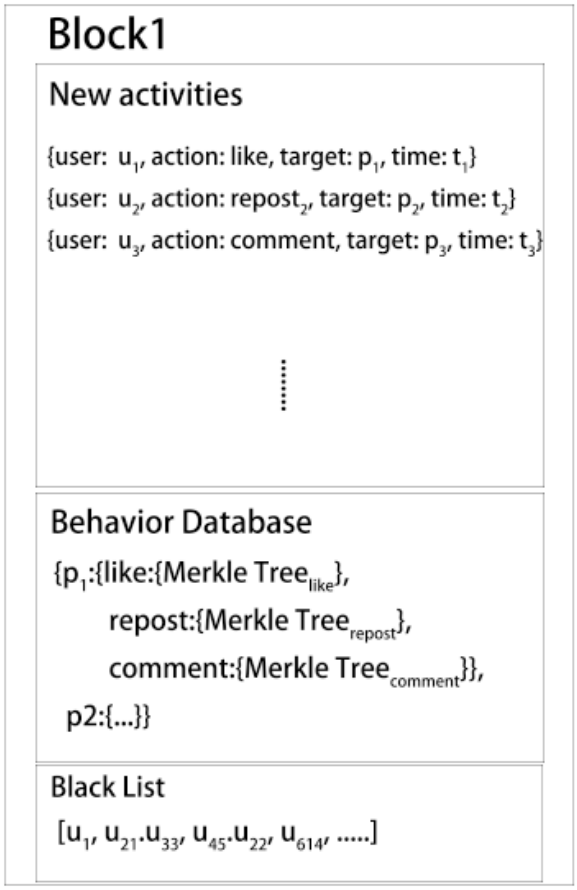

Figure 1: Block Architecture

3.2 Block Storage

Like Ethereum, each block stores a smart contract and its corresponding information. Each block stores multiple data entries (social network activities) and each entry will be stored in a format $[\texttt{user}_i, \texttt{action}_d, \texttt{target}_k, \texttt{timestamp}_t]$ 1, where `user` represents the action executor, `action` includes like, comment, repost, `target` represents the target person or figure and `timestamp` stores the creation time. An additional partial action dictionary will also be stored in each block as a distributed consensus system. Block storage capacity can later be increased to add more information, such as original social network posts or offline activity logs. This could potentially save some social network platform expenses on servers.

3.3 Blockchain Anti-spam Architecture

Attackers may directly engage in spamming to deliberately jam the whole blockchain system and cram each node's storage space. Hence, it is critical for DMSChain to implement an anti-spam penalty system to prevent errors and disruptions. There are two approaches to designing such an anti-spam mechanism, using either a single-layer blockchain solution or a double-layer blockchain solution.

The single-layer blockchain solution detects spam by searching a behavior database and running a pre-written summation program that helps create consensus in DMS Wallet. Platform maintainers then manually execute a penalty by analyzing the results of the program. Penalties are not automatically triggered and imposed.

The second solution is to use a double-layer blockchain system. A double layer blockchain system is double in storage size but retains O(n) number of blocks as in the chain of the current single-layer system. For each block creation request using this system, two blocks are created: the main block containing the same information as is in the single-layer system, and another penalty block containing information necessary to negate it. The penalty block is used as a substitute for the main block for any spam or fraud that is discovered. It is identified only by a separate (uint32) block number. Because they are mirrored opposites, separate blockchain numbers are required. For more details on the cancellation mechanism, see section 3.6 on page 9.

3.4 Shared History Database Layer

A shared history database layer provides users a transaction history of user activity as each block is created. The standard Merkle tree is used to store all blocks, instead of using a complete list of behavior records. Previous work has verified the efficiency of all operations, including insertion, query, and validation [7]. If there is a difference between two copies of the Merkle tree, one can easily spot the exact point of the first different block in the blockchain, without having to examine the whole blockchain from the very beginning. Anyone is able to validate any copy of the Merkle tree activity history database in log time with very limited computational power. This minimizes the configuration required for users to participate in DMS services.

Like Ethereum, we build multiple Merkle trees to achieve different purposes. Our current proposed design requires two Merkle trees: the Behavior Tree (BTree) stores behavior information, and the Counter Tree (CTree) stores the timestamp and trans-

action counter for that behavior. In BTree, each node stores a hash of each activity as a list of [$user_i$, $action_d$, $target_k$]. Spamming behavior could cause . In CTree, the corresponding node stores a hash list [$timestmp_t$, $counter$] including the timestamp indicating when the activity occurred and the transaction counter for that activity. We use the counter stored in CTree to identify spam or fraud. To make this anti-spam mechanism both secure and steady, super partners and distributed storage are applied at the same time. We define a super partner as any central computing resource that stores all data about activity and processes inquiries. The participating social network platforms will be required to be super partner hosts. We aim to increase the security of social network data authenticity by requiring all participating social network platforms to maintain their entire database to be monitored by everyone in the network. We will later discuss potential cheating by those participating social network platforms. We will also include some partial data in each block and expect active block owners to help do anti-spam verification.

3.5 Creation and Validation of New Blocks

The creation of new blocks can only be accomplished online or through official or other verified offline activities. In other words, offline activities must be initiated by verified organizers. They will initiate requests online once someone has attended an offline activity or bought any related goods, including handshake events, concerts, CDs, Tshirts, posters, or any other paid or free offline gatherings or purchases.

Online likes, comments, and reposts would automatically initiate requests to create blocks when users monitor social network accounts connected to various anime, comic, and idol figures. In effect, users thus engage in passive "surveillance" of these figures. (Section 4.1, page 11). Each new activity initiates creation of one block. There might be an extremely large number of simultaneous status updates so DMS must be capable of supporting a high-volume simultaneous request-processing mechanism. This will be discussed later in next section.

DMS offers a unique combination of Proof of Work (PoW)[8] and Proof of Love (PoL) for validation during block creation.

3.5.1 Proof of Work Step (direct love)

Traditional blockchain projects require massive computational resources to solve hash puzzles. By contrast, DMS defines a proof-of-work protocol that creates a block through a set of online actions. Bitcoin's current annual electricity consumption is estimated to be 48.37TWh, the equivalent of 0.22% of worldwide annual electricity consumption [9]. DMS instead aims to convert the proof-of-work concept into real world capabilities. In this sense, both offline meetup tickets and online repostings become a proof of work that actually represents real world publicity, in turn conserving computational resources.

Once an action has been completed, users are able to detect it, verify that the action has not been logged, and then send a request to create a block. This process could be automatically triggered, with accurate recording of multiple simultaneous activities that constitute direct displays of love and support for the targeted idol. We thus refer to this proof-of-work protocol as "direct love."

3.5.2 Proof of Love Step (indirect love)

DMS introduces a novel Proof-of-Love (PoL) mechanism to verify particular blocks as valid. Once a creation request is initiated, the information needs to be verified as neither spam nor any type of fraud. Currently, we define three kinds of behaviors as spam or fraud (multiple "likes" that are canceled, multiple identical or similar comments, and reposts). The anti-spam process relies on two things to determine whether the newly requested activity is spam: all N_s must be super nodes, and there must be at least $N_s + 1$ active blocks. The blocks are selected randomly from the pool of all active blocks (nodes). By doing this, we can ensure that even if all of our super dictionary nodes are attacked and represent fraud, the majority are safe and reliable.

In this part of the verification process, we ensure that each user can only create one block per social network activity. We strictly prevent spamming activities from accumulating rewards and we in fact penalize such activities, using the mechanism inherently built in our proposed blockchain system.

Any block owner who participates in the verification process will get a reward for actively helping the whole blockchain system create new blocks. There is a fixed reward for each block creation. Each block owner gets a reward proportional to the

total number of tokens he owns. Each block generates a different number of tokens depending on the type of user activity and the extent of the user's influence.

The concept behind Proof-of-Love is that if a user spends more time supporting his idol online browsing the idol's posts or other related posts or activities, he gets more reward tokens. DMS thus encourages users to commit more time to staying online to support their idol. It is designed to increase user activity and at the same time protect the DMS blockchain system from being harmed by spamming activities. DMS thus implements both direct love, directly facilitating support for the idol, as well as indirect love, indirectly helping others to do so.

3.5.3 Difference between PoS, PoW and PoL

DMS utilizes a novel PoL system by joining PoW and PoL in the process of block creation. This unique PoL mechanism is meant to more easily initiate a block creation request without relying on the extremely complicated puzzle-solving techniques traditionally needed by a PoW to qualify for new block validation. The problem with the typical PoW design in most blockchain systems is that a very high number of creation requests could block the network.

DMS's PoL essentially combines PoW and PoS. In PoL, being active in the network serves as the PoW and the direct reward from helping do the validation proportional to total wealth serves as the PoS. In DMS, each block is not considered as a single coin but rather serves as an activity recorder, making verification neither difficult nor expensive. Users are not able to manually initiate creation requests without directly operating on the social network.

DMS's PoL mechanism offers three advantages. First, it is expected to be very lightweight so as to be able to handle an extremely high volume of simultaneous requests with little delay. Second, it provides appropriate rewards for users who show ongoing support and love for an idol. Third, the architecture enables fans to easily show their support by being active online to provide valid and reliable verification to new block creations.

3.5.4 Reward Tokens

Each block contains reward tokens to pay users for their supportive actions. The number of tokens contained in each block varies and depends on two major factors: the type of action and the user's influence score. Users earn different amounts of base tokens for the three different activities involving likes, comments, or reposts: B_l, B_c and B_r. Base token amounts are determined based on the number of remaining tokens and the popularity score of the idol, as follows: $B_l = (1-T^2_{remaining}) \times (1-\frac{1}{1+e^{-I}}$, where $T_{remaining}$ is the remaining number of tokens and I is the popularity score of the idol. $B_c = 1.5B_l$ and $B_r = 2B_l$. The popularity score is the weighted score of an idol's social network account. A user's influence score is defined as follows: $I_s = \alpha \times \#followers$, where α is a constant factor. The total token gain for each action is calculated as: $T = B_i \times I_s, B_i \in \{B_l, B_c, B_r\}$.

Another way of earning tokens is to show the PoL by being active online and providing genuine and valid verifications for newly created blocks. The amount of reward tokens for each newly created block is fixed and written as RB. All participants receive some RB tokens (or portions). Suppose we have n participants and each of them has a total number of tokens T_i. Then the amount of rewarding tokens for each participant p_i is: $\frac{T_i}{\Sigma_i^n T_i} RB$.

3.5.5 Spam detections and penalties

Spam is determined to exist when the consensus of all verification nodes indicates repeated "like" activity, similar comments, or reposts of content. DMS automatically penalizes spam thus identified. Each time the user wants to use his wallet for payment or other operations, the wallet queries the network to retrieve all activities associated with that user. If one spam is found, a penalty of one reward would be imposed as a fine. If more than one spam is found, a penalty is imposed equal to all rewards associated with that user, and the user is forever blocked on DMS chain. A list of blocked users is stored in the shared history database layer and each time a block creation request is initiated, an extra step of verification is to check whether the user is on the block list, which only takes O(1) time.

In the double-layer blockchain design, the same type of penalty is imposed each time spam is detected, at which point the history database returns the previous penalty block's address and copies the information to connect to a currently created block. One repeating activity is identified as a misoperation and one copy of a reward is held back. However, if more than one repeating activity is detected, penalty blocks

are activated to cancel all rewards ever received by the spamming account user. The account will be added to the list of blocked accounts on a surveillance server.

Without an adequate anti-spam mechanism, spamming disrupts social networks and especially our blockchain system in two ways: first, it can result in an extremely high volume of requests that will block the system from further block creation; and second, it results in too many unearned rewards being unfairly given. We want to ensure that each reward corresponds to a bona fide effort to support an idol. We hope that our mechanism for block creation will incentivize fans to support their idols while also helping the idols to gain popularity.

3.6 Poison mechanism

Users need to be made aware that their online activities can generate value for the idols they support as well as for themselves; however, users are not always aware that they have received reward tokens for their online support. To raise user awareness, a timer is set to collect unclaimed tokens to encourage users to use the wallet. Collection of unclaimed tokens is also important because there is only a fixed number of tokens available to distribute. For these reasons, a mechanism is set to automatically collect unclaimed tokens in both a single-layer and double-layer design. We characterize this as a poison to kill stale rewards.

In the single-layer design, each time a user activates the wallet, all rewards generated before a defined period of time before that wallet activation would be dropped as invalid, stale rewards. Upon wallet activation, calculations would be instantly performed and the dropped invalidated tokens would be put back into the pool of total tokens available for future distribution. This is a lazy update mechanism, meaning that it is accomplished only when needed based on wallet activation and the time elapsed since rewards were earned. This means that if there are inactive users who are not opening their wallets, some tokens would fail to be recaptured to replenish the pool. To solve this, super partners could be allowed to periodically check whether users are active and to access stale tokens and put them back into the pool, without waiting for those inactive users to activate their wallets.

The double-layer design is also lightweight. No modification is needed to the double-layer blockchain network; all that is needed is to change the edges between blocks and add timestamp information in each block and in the history database. Currently,

rewards are set to be reclaimed as invalid 30 days after they are issued. A timestamp with both the date of creation of the block and the "drop dead" date for the reward to become stale and invalid is inserted into the penalty block each time a main block is created. The main block is connected to the penalty block, which is in turn connected to the next main block. The timestamp in the history database ensures that when the user activates his wallet before the drop dead date, an "antidote" can be released to retain in the wallet the rewards that are not stale. The penalty mechanism in effect drops a poison pill to cancel stale rewards when no antidote is activated. When the user activates his wallet, the system will send the antidote request to retain recent rewards. The poison will not be activated if the user has already activated his wallet.

4 DMS Network

Protocols for blockchain specifications, blockchain creation, and blockchain validation are needed to support the logic to handle all the requests to create blocks and to exchange information. In addition, however, the DMS network must be a peer-to-peer (P2P) network because typical blockchain project is expected to be decentralized, both to ensure data safety and to save computing resources. A group of centralized servers will be unable to handle all requests in the case of massive attacks, causing service to be terminated. Hence, we have designed the following network architecture.

4.1 Decentralized Surveillance Mechanism

As is discussed above, each block creation request is initialized for each new activities. However, such requests are not initialized automatically. Traditionally, we would need a set of servers refreshing once a period of time to check if there are new activities. However, in DMS, such setting could be extremely expensive and is not adpatable to scale easily. Once the network gets bigger and the idols on chain become more popular, the server would need to expand exponentially. Therefore, we will introduce a peer report mechanism that would save the high expense of servers.

Each new activity initiates a new block creation request, but does not do so automatically, but rather does so through a peer report mechanism. This dispenses with the need for and high expense of a set of servers to periodically refresh to check for new activity. This also solves the problem of scalability: once the network expands and the idols on the chain become more popular, the server would need to expand

exponentially.

A widget or extension must be install in the user's phone or browser to support this mechanism. There are two ways to initiate block creation requests. One occurs when a user reacts to an idol's post, that user's widget or extension checks to see if there are new posts and yet reported to the chain reactions. This would initiate requests of all the new activities to the chain. However, that particular user might not have installed the necessary widget or extension. The alternative is for the widget or extension installed by other users to access the Internet and when traffic is freed up, automatically iterate through the participating idol's social network pages to identify and report any missing activities to the chain.

Although at the very beginning of service, some surveillance servers should be used to help the block creation process, the P2P network is ideal with a mature, steady network to achieve completely decentralized service.

4.2 Super Partner

The super partner is a core concept in the DMS blockchain system to form the basis for a steady network. Super partners are those users who store the whole database providing a steady connection for validation and are considered to be extremely creditworthy. Traditionally, super nodes in Bitcoin are not trusted since there is still a significant risk of cheating. By contrast, super partners offer have good credit ratings, are trusted, and are big companies monitored by other super partners. A company user will lose super partner status of inconsistencies occur.

4.3 Distributed Hash Table

A hash table serves as a map to locate any node in the DMS network. DMS uses a distributed hash table, storing information at a large network node. which in the central hash table, we store the information to some big node in the network, not the whole. The setup of the distributed hash table is like a typical Kademlia-like distributed hash table[10], as has been used in many previous blockchain projects. This means that much less data need be stored by super partners in a core hash table, thus taking full advantage of the blockchain architecture to save space.

5 DMS Shared History Database

Key information about previous user activities must be retained in a database, mainly to detect spam. Ideally, the original text of comments and posts on the social network could be stored, but this is as yet not feasible. Nevertheless, we suggest how this could be built as we continue to explore more options.

A database of shared history will obviously grow as more blocks are created and block size will also grow. For that reason, each block should be kept as small as possible, without risking data loss. While there is a set of super nodes in the network, we store some partial data in each block. Each set consists of 500 blocks of information and each node must store 1/10 of the total information in those 500 blocks. Moreover, each block of information is not stored more than 50 times (1/10 * 500). In this sense, if we select enough blocks randomly, we can still guarantee that we query the whole database. We will also explain this part in the future.

6 DMS Wallet

DMS Wallet is the hub for DMS users to manage reward tokens. The wallet app allows users to access all owned tokens, which the user can spend or transfer. The wallet shows users details about their rewards. DMS wallet needs to be activated before collecting the rewards. Users need to verify their identities on different social network platforms using the APIs provided by each platform. Such setting ensures that everyone can still keep their anonymity while accessing and collecting their rewards. No cross-platform account connections would be available to other users. Then, a surveillance widget or extension is required to be installed since it is the basis for the entire blockchain network. Everyone contributes some computational resources and in return receives steady service.

After activating their wallets through other social media platforms by verifying their identities, users are able to collect their rewards. Each wallet is an address that can be used to make purchases or transfers. Such payments are supported via a traditional Proof of Stake (PoS)[11] method and are also validated by other active users (supported by the transaction tree). We will further develop the wallet in future work.

7 Conclusion

In this white paper, we propose DMS, a scalable multi-blockchain architecture that features anti-spam and timer mechanisms. The proposed DMS blockchain system suggests the possibility of other secure blockchain applications that could be applied in many other areas. We welcome input as to how to build on the design we suggest here.

Our key contribution in this white paper is the introduction of a Proof of Love (PoL) validation mechanism, an anti-spam mechanism and a poison mechanism to recapture unclaimed rewards. This DMS blockchain proposal targets the anime, comic, and idol industry to grow and prosper.

In next version of our technical white paper, we will finish the design of the DMS distributed database and in-chain payments and we will also add more security features to ensure the safety of the DMS blockchain system.

References

[1] M. Swan, *Blockchain: Blueprint for a new economy.* " O'Reilly Media, Inc.", 2015.

[2] G. Fox, "Peer-to-peer networks," *Computing in Science & Engineering*, vol. 3, no. 3, pp. 75–77, 2001.

[3] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai, "Uncovering social network sybils in the wild," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 1, p. 2, 2014.

[4] N. B. Ellison *et al.*, "Social network sites: Definition, history, and scholarship," *Journal of computer-mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

[5] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?," in *Proceedings of the 19th international conference on World wide web*, pp. 591–600, ACM, 2010.

[6] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, 2014.

[7] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Conference on the Theory and Application of Cryptographic Techniques*, pp. 369–378, Springer, 1987.

[8] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[9] "Bitcoin energy consumption index."

[10] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *International Workshop on Peer-to-Peer Systems*, pp. 53–65, Springer, 2002.

[11] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, 2014.