Ternary Funding and a Novel Type of Non-Fungible Token: A New Way to Incentivise Public Goods

AJESIROO

ajesiroo@protonmail.com

July 14, 2021

ABSTRACT

We introduce the concept of ternary funding and joint tokens – a novel type of NFT that is procedurally generated from a pool of validators. To meet today's need, we primarily focus on an implementation on Ethereum, but briefly explore how these concepts can be applied to a theoretical distributed ledger under an anarchistic model.

1 INTRODUCTION

1.1 Structure

This paper is divided into two main sections. Section 2 describes a funding model for public goods that is implementable today using smart contracts on Ethereum.[1] Section 3 describes how the principals in the former can be adapted to a theoretical distributed ledger under an anarchistic model.[2]

1.2 Not-For-Profit

All mechanisms described herein are not-for-profit. There is no native token and the atypical staking pool described in Section 2.3 has no direct reward for validators.[3] In order to eliminate the volatility of Ether, a decentralised stablecoin is used for both the pool outlined in Section 2.2 and Section 2.3.

1.3 Public Goods

The term *public good* in the context of this paper refers to goods that are non-excludable, non-rivalrous and include the consideration of externalities.[4] The example provided in Section 1.4 is a public good in the form of open data.

1.4 Primary Example

The primary example of a public good that is referenced within this paper is the findings of a team of clinical researchers conducting a small study on the efficacy of an immunotherapy for paragangliomas. This is used only for illustrative purposes, and the mechanisms described herein can be applied to any public good that meets the criteria in Section 1.3. The aforementioned example

[1] The author acknowledges the carbon footprint of Ethereum as a byproduct of proof-of-work. Nevertheless, Ethereum, and specific EVM-compatible rollups described in later sections, remain the most suitable candidates for deployment given their ubiquity and level of security without sacrificing decentralisation. The high level of energy consumption is also, of course, a temporary problem as the transition to proof-of-stake is already underway, with the Beacon Chain launched and its subsequent merge with the Mainnet on track.

[2] The assumptions of the anarchistic model are provided in the overview of Section 3.

[3] The social incentives for participating in such a staking pool are described in later subsections.

[4] See https://otherinter.net/research/positive-sum-worlds/ for discussion on externalities in the context of public goods.

was chosen as it is small in scale and describes research for an underserved cohort that receives little funding under traditional models.

2 ADDRESSING THE CURRENT NEED

There is a need to fund public goods that provide benefit to underserved communities such as those described in Section 1.4. That is why there is particular emphasis within this paper on what is implementable today, rather than what is possible on the theoretical distributed ledger described in Section 3.

2.1 Ternary Funding

We introduce a funding mechanism consisting of three primary stages:[5]

- Population of a staking pool,[6] of which, a minimum number of validators are needed to progress to subsequent stages
- Population of a donation pool, or as described in Section 2.4, the generation of a joint token
- Attestation of the outcome by a random selection of validators, with release of the donation pool or joint token upon delivery of the public good

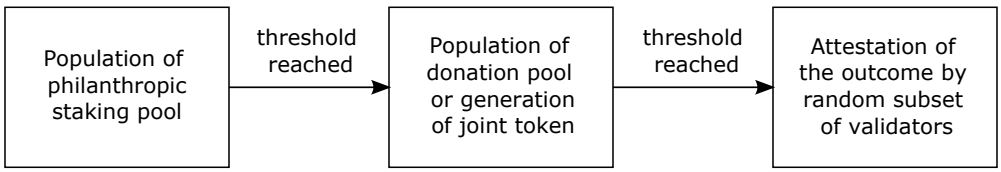

Fig. 1

The cycle is renewed for each funding round and a DAO is not needed to manage subsequent rounds or changes to parameters (see Section 2.3.7).

Critically, the staking pool must be populated before the donation pool. This is for several reasons:

- While validators that are randomly chosen to attest to the outcome are pseudonymous, all validators are initially identified (see Section 2.3). This, together with a minimum threshold of validators being reached, provides a degree of legitimacy to the funding round, which is the mechanism that incentivises contributions to the donation pool.
- The threshold of validators being reached is an implicit agreement of the parameters of the funding round, most notably, acceptance of the party developing the public good, and the specific criteria of what constitutes the materialisation of the public good (see Section 2.3.6).

As the attestation of the staking pool is the mechanism that either releases or reverses the funds in the donation pool, the amount in the donation pool is capped in order to be within proportion of the staking pool. This is to prevent an attack where the cost of bribing validators, or validators otherwise colluding, is lower than the potential payout of the donation pool (see Section 2.3.2). As also explained in Section 2.3.2, the penalty for attesting outside of the three-fourths majority is the slashing of the entire stake.

[5]The term "ternary" is a reference to the fact that there are three stages. It can alternatively be referred to as "sequent funding" as the stages generally resemble sequent logic.

[6]"Staking pool" in this context should not be confused with an Ethereum PoS staking pool. Participants in the pool are the validators themselves, rather than users delegating their stake.

Finally, while the contracts are ideally deployed to layer 1, gas fees are often prohibitively expensive, and EVM-compatible rollups must also be considered. At present, the only EVM-compatible layer 2 option that is *deployment-ready* (and provides the full security of the Mainnet) are optimistic rollups.

2.2 Donation Pool

Once the threshold of validators in the staking pool described in Section 2.3 is reached, contributors can populate this pool up until the cap outlined in Section 2.1. It is important to note that contributors are able to participate in both pools.[7]

2.3 Philanthropic Staking Pool

This is a novel decentralised oracle that determines whether the stablecoins locked in the donation pool, or the newly minted joint token described in Section 2.4, should be released or returned. In the case of a joint token, the token is simply burned.

Unlike other staking pools, there is no cryptoeconomic incentive to become a validator. The staked tokens are never converted into interest-bearing tokens[8] and the validator does not receive a direct reward once the attestation is undertaken. On the other hand, there are costs involved to become a validator, thus the staking pool is itself a donation pool in the vein described in Section 2.2.

The staking pool is, in effect, donating the following:

- The opportunity cost represented by the risk-free rate (R_f)
- A small but theoretical risk that an honest validator's stake is slashed if it does not attest in line with the three-fourths majority
- The gas fees involved in interacting with the staking contract

It is important to articulate the motivations for participating in such a pool. As described in the overview of Section 2.1, the participants of this pool are not anonymous. The reason for this is threefold. To prevent various attacks described in later subsections, to allow the donation pool to gauge the quality of the funding round, and most pertinently, to provide a social incentive for validators to participate. Validators can publicly post the transaction hash once they have staked the required amount. Thus the incentive is similar to other philanthropic contributions. The main difference with a simple donation, however, is that an honest validator can expect to receive their staked tokens back,[9] minus R_f and gas fees.

2.3.1 Sybil Resistance. The inherent properties of the staking pool mitigates this attack. Validators are not anonymous (see Section 2.1) and only during attestation is a random selection of validators pseudonymous (see Section 2.3.5).

2.3.2 Collusion. This is a much more complex problem [1, 2]. Nevertheless, we show how this is unlikely to be successful given the inherent design of the various pools. As explained in the overview of Section 2.1, the total amount locked in the donation pool is capped in order to be kept in proportion with the staking pool. In the case of bribing validators, a bad actor would need to ensure that the total expenditure is less than the potential payout of the donation pool. As the random selection of validators that are chosen to attest does not occur until the final stage, and

[7] The reason why the pools are not simply merged into a single pool is because some may want to donate without the additional burden of attesting.

[8] The version the author is working to implement will not transfer tokens to a lending platform due to the increased counterparty and smart contract risk. Other implementations could use a third-party lending platform so long as all interest earned is provided to the party delivering the public good (after attestation) rather than validators.

[9] Provided they attest in line with the three-fourths majority.

their addresses are obfuscated via zk-SNARKs (see Section 2.3.5), the bad actor would have to bribe the vast majority of validators in order to confidently meet the three-fourths needed to attest to a malicious outcome. If the bad actor was not able to persuade enough validators, the entire stake of all compromised validators is lost during attestation. This makes collusion prohibitively expensive.

Even in the most extreme scenario where the originator of a funding round (who can remain anonymous) is malicious and is associated with all validators in the staking pool, the donation pool would most likely not reach the minimum threshold required to progress to the final stage. As outlined in Section 2.1, the staking pool is populated before the donation pool precisely to allow participants in the latter to the assess the quality of validators and the fund originator. An anonymous fund originator as opposed to a known party (see example provided in Section 1.4) is unlikely to receive enough funding to allow progression to the final stage. When the minimum threshold in either of the two pools is not reached, all funds are reversed. This also addresses validators simply colluding out of some common interest. In this circumstance, the donation pool can be expected to not receive the minimum threshold if most participants in the staking pool are not trusted.

In the case of a newly minted non-fungible joint token in lieu of donation pool (described in Section 2.4), the joint token would have no intrinsic value but still incur the costs outlined in Section 2.3.

2.3.3 Interval Before Attestation. There is a time-based interval between the delivery of the public good and the attestation of the outcome. This is to provide enough time for validators to decide whether the criteria set at the beginning of the funding round have been met. For a typical funding round, this might range from weeks to months, depending on the nature of the funding. For the example outlined in Section 1.4, the interval should be roughly equivalent to the time required for peer review. It is important to note that the actual subset of validators that are randomly chosen to attest are not selected at the beginning of this time-period, but at the end.

2.3.4 Cycling of Validators. A random set of validators are selected to attest to the outcome. This occurs after the interval described in Section 2.3.3. As an anti-collusion measure, the window of time the validator has to attest is short, and if they are not available, a different validator is selected until enough validators have attested. If the required number of validators do not attest, all funds in the staking pool and donation pool are reverted, and in the case of a joint token, the token is burned.[10]

2.3.5 Attestation. Once a random set of validators are chosen,[11] a key is generated and each attestor submits a transaction containing the zk-SNARK of this key together with their attestation.[12] The zk-SNARK is needed to ensure only validators that are chosen to attest can call the necessary function. Once enough verified attestations occur, the stablecoins secured by the staking contract are either released or reverted. In the case of a joint token in lieu of a donation pool, the token is either released or burned.

[10] The funding round is considered cancelled, and all funds are returned to their originating addresses.
[11] For example, 16 out of a total pool of 64 validators.
[12] A zk-SNARK for each voting outcome (0,1) should also be considered, although this will use more gas.

The decision can be shown as:

$$\sum_{i=1}^{v} attestation_i \tag{1}$$

where v is the total amount of validators chosen to attest.

It is also theoretically possible[13] to remove the need for an intermediary attestation contract which verifies the zk-SNARK of the key through the use of *private* zk-rollups[14] – that is, zk-rollups that use recursive proofs [3]. In this case only a private transaction would need to be sent to the staking contract, and a proof of the rollup verifies the proof of the underlying transaction.

For a given zk-SNARK (G, P, V), where G is the generator algorithm that outputs the proving key G_p and verification key G_v, the output of recursive G is (G'_p, G'_v).

2.3.6 Subjectivity of the Outcome. Validators should be encouraged to only stake in pools where the funding originator has provided a short and highly specific criteria on what constitutes the delivery of the public good. This is to make it as easy as possible to reach the three-fourths consensus during attestation. It is important to reiterate from Section 2.1 that all funds in the staking pool are reversed if the required threshold of validators is not reached to progress to the next stage, therefore it would require the minimum threshold of validators to make this oversight for the funding round to progress.

There will nevertheless be an inherent degree of subjectivity given the nature of the decentralised oracle. Those that attested in line with the three-fourths majority will always receive their stake back, regardless of the decision on the outcome. This also applies for circumstances where no three-fourths majority is reached.

2.3.7 Omission of a DAO. While a DAO can be used to manage the parameters of each funding round, it is not necessary. A new lightweight contract can be deployed for each successive funding round and an interface can be defined that links to a common set of contracts in order save on the gas fees of deployment. The ERC-721 standard only requires that each address and `tokenId` form a unique pair, therefore it does not matter if there no continuity in the `tokenId` so long as a different contract address is used. The advantage of this approach is that no governance token is needed, and the acceptance of the new parameters is contingent on whether or not the new contract receives the necessary threshold of validators.[15] It also negates the ability of a party to spam trivial funding rounds.

2.3.8 Stablecoin. A decentralised stablecoin is used for both the staking pool and the donation pool. The primary requirements are that it is sufficiently decentralised and secure. At present, Dai is the only candidate that meets both these criteria.

2.3.9 Multiple Known Accounts. As previously mentioned, the staking pool relies on identity. The preferred way to achieve this is to adopt a proof of personhood system.[16] Given the infancy of the major projects in this space however, a temporary workaround can be described as multiple known accounts (MKA).

This is simply a validator posting the transaction hash of their stake to *multiple* major social media platforms. It's important to note that most individuals that have a degree of reputation are

[13] There are no projects currently deployed to the Mainnet that enable this.

[14] An example of a project currently working on this problem is Aztec.

[15] This also addresses the funding allocation problem.

[16] Proof of Humanity and BrightID are currently the most feasible options.

not formally verified by the platform itself. It is up to participants in the next stage, the donation pool, to assess the quality of validators and decide if they want to participate in the pool.

Crucially, the transaction hash needs to be posted to multiple platforms. This is to provide a degree a decentralisation rather than relying on a single platform.

2.4 Joint Token

The donation pool described in Section 2.2 can be replaced entirely with a novel type of non-fungible token introduced herein as a joint token (JT).[17] The token is generated pseudorandomly based on the addresses of all validators, and can only be minted once the staking pool is populated. Validators then decide whether to release or burn the token, in the way described in the overview of Section 2.3.

While the algorithm for generating the token is detailed in Section 2.4.4, it can be summarised as a mapping of each address to a colour, which then collectively form an n^2 grid representing all validators.

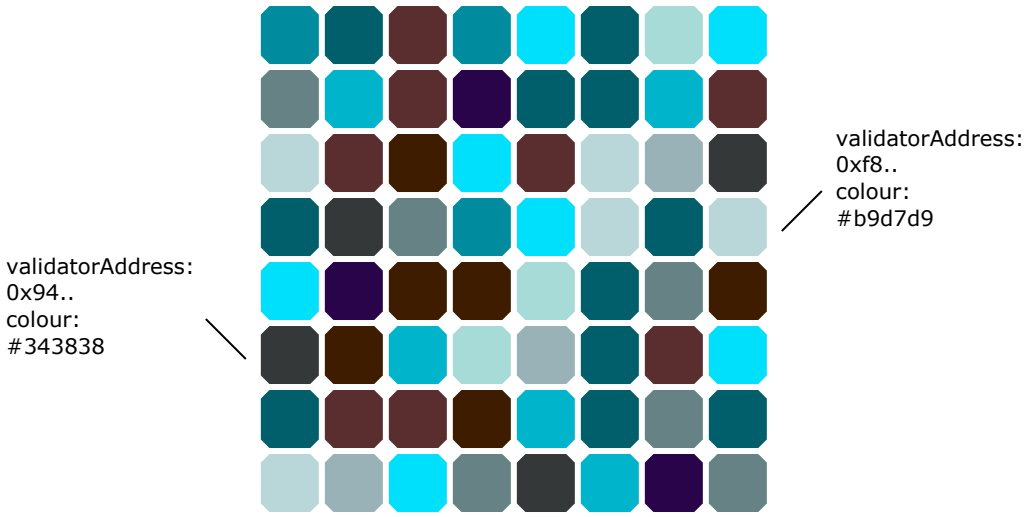

Fig. 2

It is important to note that all data relating to the token is stored on-chain. It primarily consists of an array of structs containing each address and colour pair. The figure shown above is simply the parsing of this data structure off-chain. The token conforms to the ERC-721 standard, and there is *no* image URI contained in the metadata schema, as only an array of structs containing each pair is necessary to develop a visual representation off-chain.

More complex mappings beyond simple colours can be used, however gas fees quickly become a constraining factor, especially for funding rounds with many validators.

[17] A hybrid funding round containing both a donation pool and joint token is technically possible.

2.4.1 Value. In the event of a successful funding round, and the token is released to the party that delivered the public good, the token can be expected to have a degree of intrinsic value based on the following:

- All validators that contributed, and consequently paid the costs outlined in Section 2.3, are represented individually on the n^2 grid. For each address (and hence identity),[18] it is possible to ascertain their exact position on the grid, given the data structure stored within the token.
- The token is recognition of not only the party that delivered the public good, but of validators who themselves made contributions. As outlined in Section 2.3, there is no cryptoeconomic incentive to become a validator, and this is by design.

Once the token is released, the receiving party can opt to either keep it or trade it.

2.4.2 JT-Specific Incentives. Similar to the staking pool described in Section 2.3, the incentive to generate a joint token is social rather than cryptoeconomic. The validator can publicly post their transaction hash, in the same way an entity can publicly announce a philanthropic contribution.[19] As the staking pool can be capped, an additional incentive is scarcity – If the grid is a mapping of 64 validators,[20] then a given address can only be associated with the token if the contribution is made before the cap is reached.

2.4.3 Preference of Funding Vehicle. As there are no longer funds risked in the donation pool described in Section 2.2, a joint token should be the *preferred* funding mechanism for most rounds.

2.4.4 Procedural Generation. The following is a truncated algorithm in order to illustrate the key components needed to generate the joint token.

We first define a struct that that will hold the address and colour pair.

```
struct Validator {
    address validatorAddress;
    string colour;
}
```

The string `colour` corresponds to the hexadecimal code that is ultimately selected, e.g `#008b46`. For efficiency purposes, this can be represented as a set of integers that form the RGB equivalent, i.e. `0, 139, 70`. In order to keep the truncated algorithm simple however, we will use a predefined palette that is represented as an array of strings.

We then initialise an integer representing the total number of validators, and an array of structs, validator addresses and a predefined palette.

```
uint validatorCount;

Validator[] coloursArray;
address[] validators; // [0x7a.., 0x6B.., 0xb77.., ..]
string[] palette; // ["#ece5ce", "#424242", "#78c0a8", ..]
```

[18] As previously mentioned, validators are identified.
[19] In some jurisdictions, tax benefits may also apply.
[20] This number is provided as an example. The value of this parameter should be highly dependant on the funding round.

We define a function that pseudorandomly selects a colour based on the validator address and the current block timestamp. Note that a third-party oracle is not needed as we do not require a more secure source of randomness for this specific application. A hash function that accepts the validator address and `block.timestamp` is sufficient.

```
function rand(address _validator) internal view returns(uint256) {
   uint256 seed = uint256(keccak256(abi.encodePacked(_validator, block.timestamp)));
return seed % validatorCount;
}
```

We populate the array of structs that contains each pair.

```
function setColoursArray() internal {
    for(uint i=0; i<validatorCount; i++){
        coloursArray.push(
            Validator({
                validatorAddress: validators[i],
                colour: palette[rand(validators[i])]
            })
        );
    }
}
```

We define a function that gets all pairs.

```
function getColoursArray() public view returns (Validator[] memory){
    return coloursArray;
}
```

Table 1. Output

Index	validatorAddress	colour
0	0x3a..	#78c0a8
1	0xc20..	#ece5ce
..	..	..

As each funding round is associated with a single token, `coloursArray` is the token itself, not an array of tokens. `coloursArray` needs to be transferred to an address, per the ERC-721 standard, and the `tokenId` is the index of the array containing all `coloursArray`'s.

Once the array is parsed off-chain, we can build the grid that can be presented in the browser.[21]

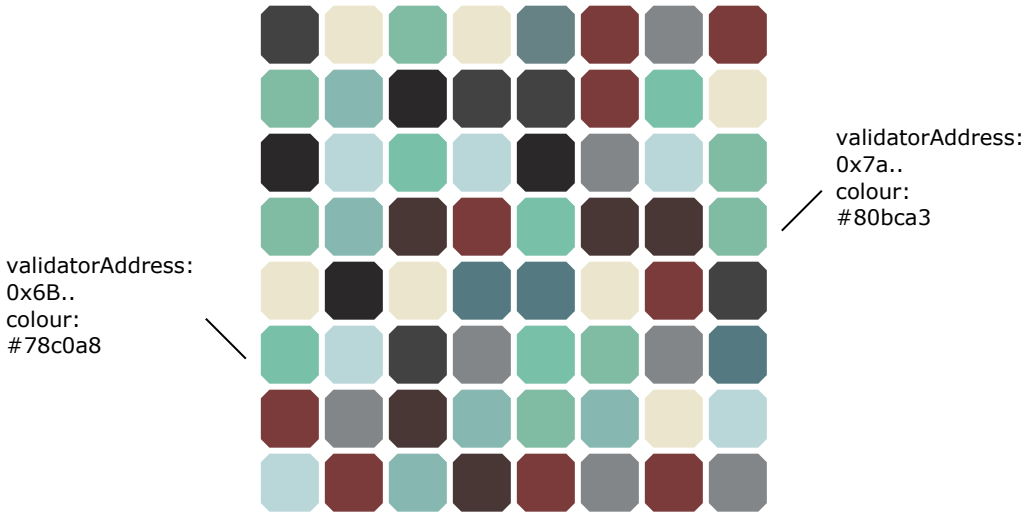

Fig. 3

2.4.5 Colour Permutations. For simplicity, we assume a funding round of 64 validators v and a palette of 64 colours c. When colours are able to be reused in the same funding round, we observe $c^v = 3.94 \times 10^{115}$ permutations. In the case that colours can only be selected once, we observe $\frac{c!}{(c-v)!}$ $= 1.27 \times 10^{89}$ permutations.

3 UNDER AN ANARCHISTIC MODEL

This section describes how some of the concepts from the former can be adapted to a theoretical, smart-contract enabled distributed ledger under an anarchistic model.[22] It is relatively brief as there is little point in being overly prescriptive for a society that doesn't exist.

3.1 Assumptions of the Anarchistic Model

It is important to provide some assumptions of this model, in order to provide clarity for later subsections.

- We assume a stateless, classless, moneyless society, with the elimination of all hierarchies.
- Hierarchies that remain for practical purposes are scrutinised, until means for their elimination can be found.
- Basic material needs are provided unconditionally, even to those who either fail to come to an agreement with the commune or fail to uphold their obligations to the commune.
- Subjective needs,[23] also known as luxuries, are needs in excess of material needs [4].
- The concepts described in later subsections refer to the voluntary tokenisation of subjective needs only. Material needs cannot be tokenised, as they are already provided to all unconditionally.

[21] ENS name can also be displayed if reverse resolution is enabled.
[22] In some ways, the model can be described as "distributed contract anarchism", and while outside the scope of this paper, distributed contracts can theoretically be used for other communal contracts outside of mutual tokens introduced herein.
[23] The term was first coined on the channel *Angie Speaks*.

- The tokenisation of goods described in later subsections do not negate subjective needs being distributed via other means, such as amongst groups that share a common interest, or simply being given away.

Unlike in Section 2, there is only one pool – a staking pool for the voluntary tokenisation of subjective needs. The underlying goods or services that collectively form a token can either be wholly digital or physical, and upon successful attestation, all items are transferred to the receiving party (see Section 3.4.1). These tokens are distinct from money, as they represent underlying goods or services.

The purpose of locking the token until attestation occurs, is similar to that of the joint token described in Section 2.4 – that is, to incentivise certain outcomes. Even under an anarchistic model, some preferences may go unmet – for instance, research into therapies for a rare disease.[24] The purpose of the mutual tokens described herein is to signal those preferences.

3.2 Simple Mutual Tokens

These are tokens similar to the joint tokens described in Section 2.4, however because we are now assuming a theoretical distributed ledger with dramatic improvements in scalability compared to current blockchains, we can replace the basic colour mapping, with large payloads of data.[25]

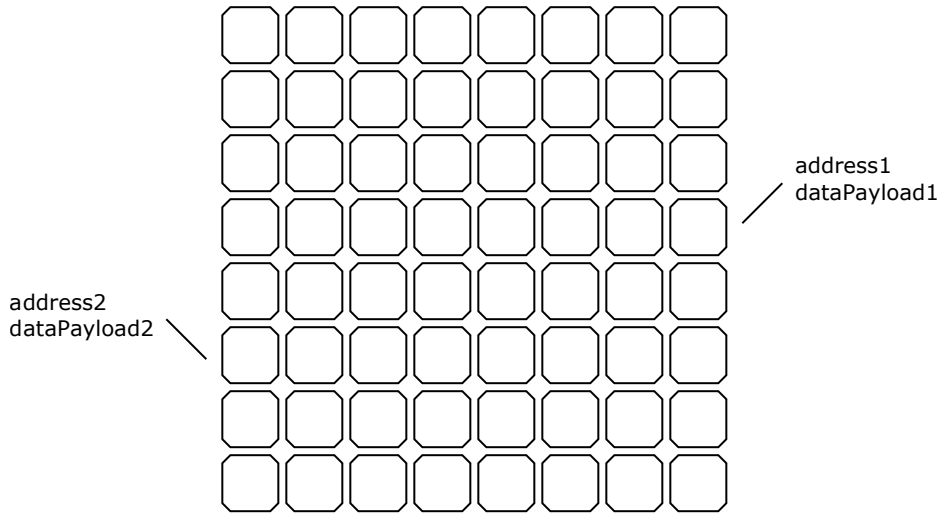

Fig. 4

Each payload of data contained within the array is a digital subjective need. In the case the data is not encrypted, value is derived from originality.

[24]While under anarchism it is assumed that work towards most medical research would be voluntarily undertaken without needing additional incentives, it is clear that some disease processes affect only an extremely small amount of people, and an additional signalling mechanism is sometimes needed.

[25]Unlike joint tokens, the data is generally provided by each validator rather than procedurally generated.

3.3 Complex Mutual Tokens

These are more cumbersome as each item in our array now corresponds to physical items rather than a payload of data. Nevertheless, as described in Section 3.1, it would be unproductive to stake material needs, as they are already given to all within the commune unconditionally.

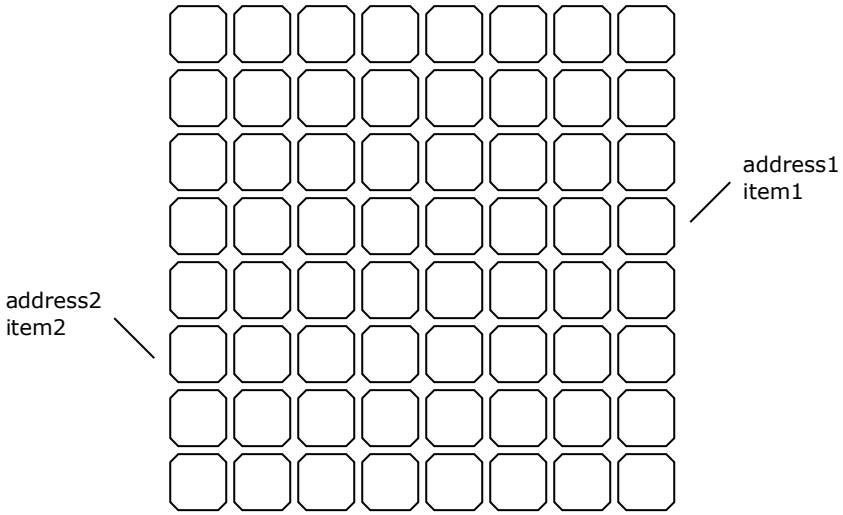

Fig. 5

Each item in the array is essentially a promise to provide the subjective need if consensus is reached amongst the pool to release all items within the pool. The items are not held in any kind of escrow and there are no checks on quality. It is effectively a form of mutual aid. An entity develops a good that otherwise was lacking, and the staking pool reaches a consensus on whether or not to release the token. All party's basic needs are already provided, and subjective needs can still be acquired via other means. Mutual tokens are simply an additional mechanism to incentivise the development of certain goods.

3.4 Other Considerations

3.4.1 Transfer. In the case of simple mutual tokens, the items are wholly digital, therefore the smart contract automatically transfers the items in the array to the entity once consensus is reached.

In the case of complex mutual tokens, the items are physical, and rely on the principals of mutual aid. All information on our theoretical distributed ledger is publicly available, and if the items staked are not delivered, this is visible to other members of the commune. The public availability of this information is also the mechanism that addresses the nothing-at-stake problem during attestation.

3.4.2 Accumulation. If an entity were to accumulate a quantity of mutual tokens that could lead to the development of a hierarchy, it is important to note that the ledger is public.[26] Furthermore, the tokens do not have the same velocity of money, as they represent underlying goods or services. Non-exploitable personal property is recognised in virtually all major tendencies of anarchism [5–7] from mutualism through to anarcho-communism.[27] Mutual tokens are simply the voluntary tokenisation of these goods on a public ledger.[28]

REFERENCES

[1] Vitalik Buterin. 2019. *On Collusion.* Retrieved Jun 17, 2021 from https://vitalik.ca/general/2019/04/03/collusion.html

[2] Siddarth Divya, Ivliev Sergey, Siri Santiago and Berman Paula. 2020. *Who Watches the Watchmen? A Review of Subjective Approaches for Sybil-Resistance in Proof of Personhood Protocols.* In Frontiers in Blockchain, Volume 3. DOI: https://doi.org/10.3389/fbloc.2020.590171

[3] Juan A. Garay and Rosario Gennaro. 2014. *Advances in Cryptology – CRYPTO 2014.* 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II. Germany, Springer Berlin Heidelberg.

[4] Abraham Maslow. 1943. *A Theory of Human Motivation.* Psychological Review, 50, 370-396. Retrieved Jun 17, 2021 from https://psychclassics.yorku.ca/Maslow/motivation.htm

[5] Peter Kropotkin. 1906. *The Conquest of Bread.* Retrieved Jun 17, 2021 from https://libcom.org/library/the-conquest-of-bread-peter-kropotkin

[6] Pierre-Joseph Proudhon. 1840. *What is Property? An Inquiry into the Principle of Right and of Government.* Retrieved Jun 17, 2021 from http://etext.lib.virginia.edu/toc/modeng/public/ProProp.html

[7] Mikhail Bakunin. 1873. *Statism and Anarchy.* Retrieved Jun 17, 2021 from https://libcom.org/library/statism-anarchy-mikhail-bakunin

[26] It is up to the commune to decide how to deal with property that is being exploited.

[27] Even in Kropotkin's interpretation of property – where goods are the product of everything that came before it, and therefore owned by everybody – access to non-basic needs is seen as essential.

[28] As it's relatively trivial to implement mutual credit systems on distributed ledgers, it is technically possible to use mutual tokens as part of a mutual credit system. However the author is not in favour of Proudhonism, and views mutual credit systems as overly rigid. Nevertheless, the author is in favour of these systems (e.g Circles UBI) under the status quo.