The Governance Problem in Distributed Ledgers: An Analysis Focusing on Tezos

Yussef Soudan

4th Year Project Report
Computer Science
School of Informatics
University of Edinburgh

2021

Abstract

With blockchain platforms evolving to adapt to changing user needs, governance mechanisms are needed to guide that evolution and to avoid community divisions. Off-chain and on-chain governance mechanisms run the risk of community division when adopting or rejecting updates to their ledgers, given each stakeholder's choice to move to the updated chain or to stay on the original chain. However, self-amending blockchains do not allow users not to update their software, in order to avoid hard forks. In this work, we provide an analysis of Tezos, a self-amending on-chain governed blockchain, to examine its governance attributes and any risks it poses to community division. We show that Tezos offers some desirable governance properties, such as universal and individual verifiability, but lacks others, such as private ballots and incentivised participation. We also prove that the Tezos governance system is not Pareto efficient, and that in the latest state of supply distribution it is feasible for community division to be the most optimal outcome for stakeholders.

Acknowledgements

I would like to thank my supervisor, Professor Aggelos Kiayias, for his constant guidance and support throughout the academic year. I would also like to thank Dr Nida Khan for her prompt communication in clarifying her application of Nash equilibria to blockchain governance.

Contents

Chapter 1

Introduction

1.1 Motivation

Following the founding of Bitcoin [1] in 2011, cryptocurrencies and other blockchain platforms have tremendously risen in popularity. Unlike centralised organisations, which are governed by a select few, blockchain platforms can be governed in a decentralised fashion by the different actors in these platforms. The decentralised nature of blockchains has been central to their appeal; however, it has also introduced new challenges. Blockchain platforms, like other organisations, try to adapt and adjust to their stakeholders' needs and preferences. With different actors present whose preferences might not always align, governance problems arise and the risk of division within the community increases.

Different governing mechanisms exist, depending on the platform. Off-chain governance is the most centralised of such mechanisms with the core developers making most of the decisions. On-chain governance is achieved via on-chain voting mechanisms, which can be more transparent and inclusive than off-chain governance. In both of these mechanisms, community division can take place when a backward-incompatible update is adopted, where some stakeholders choose to stay on the original chain and others choose to upgrade to the updated chain, dividing the community into two. In the most general sense, this is known as a hard fork. Such divisions can fragment the community and its resources, and reduce the overall value of the platform.

Recently, Tezos [2] has been founded as a self-amending blockchain, which can adapt to changing preferences without hard forks. That is, when an update to the existing protocol is adopted, users cannot choose not to upgrade. We are interested in analysing the different aspects of the Tezos governance system and in analysing how it can avoid community divisions.

1.2 Contribution

The contributions of this work can be summarised as follows:

1. Examination of some of the existing blockchain governance systems and their failures.
2. Examination of the most prominent governance models proposed in the literature.
3. Derivation of five desired properties of blockchain governance from real-world governance examples, governance models, social choice theory and game theory.
4. Outline of the Tezos governance system.
5. Evaluation and analysis of Tezos with respect to the derived properties of blockchain governance.

1.3 Thesis Structure

- *Chapter 2: Background.* We introduce blockchain governance, and provide an overview of existing blockchain governance systems and proposed governance models.
- *Chapter 3: Desired Properties of Blockchain Governance.* We derive five desired properties for blockchain governance from real-world examples, governance models, social choice theory and game theory.
- *Chapter 4: Overview of Tezos.* We outline the consensus protocol of Tezos and its governance system.
- *Chapter 5: Analysis of Tezos.* We evaluate and analyse Tezos with respect to the properties derived in Chapter 3.

Chapter 2

Background

2.1 An Overview of Blockchains

The word 'blockchain' became truly mainstream with the founding of Bitcoin [1], the world's largest cryptocurrency by market-cap [3], in 2009. In this work, we define a blockchain as a distributed, decentralised, tamper-proof ledger. The ledger records a set of transactions, which are verified through a decentralised consensus process among trust-less validators before being attached to the chain. The following is a bottom-up breakdown of the structural components of such a ledger:

- *Transactions*. A transaction specifies a monetary value, the address of the sender, the address of the receiver and a transaction fee. Transactions are proposed by blockchain nodes, or users.
- *Blocks*. Each block is composed of a block header and a list of transactions. The block header contains metadata, among which is a hash pointer and the hash of a binary hash tree.
- *Hash Pointers*. A hash pointer in a certain block specifies the hash of the content of the previous block [4].
- *Binary Hash Trees*. A hash tree is a tree in built using hash pointers (instead of regular pointers), where transactions of the block are stored at the leaf nodes of the tree. As we go up the tree, each non-leaf node contains the hash value of the concatenation of its child nodes. An example of a binary hash tree is a Merkle tree [5].
- *Blockchain*. We can think of a blockchain as a growing linked list of blocks, except hash pointers are used to link the list. A simplified image of the overall structure from [6] is included in Figure 2.1.

The extensive use of hash pointers to link the various structural components allows the blockchain to be *immutable*, or tamper-resistant. Since a copy of the blockchain is contained on multiple computers that form the blockchain network, an attempt to alter the stored data will be detected and ignored as an illegitimate modification. Immutability comes with certain challenges, however, as we will discuss later.

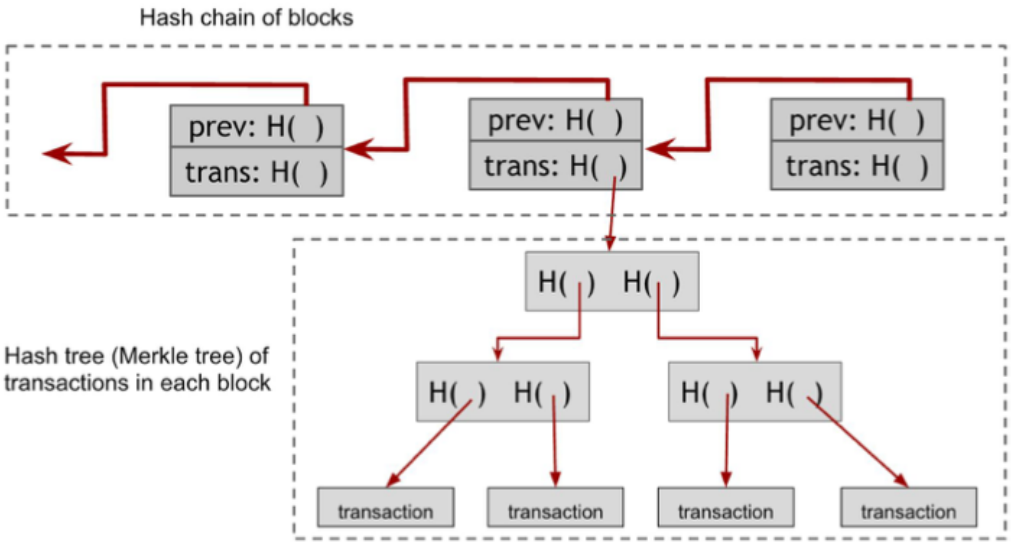

Figure 2.1: An example of a blockchain data structure where the transactions are included in the block and the block points to a root of a Merkle tree.

2.2 The Blockchain Network

A blockchain is not maintained by a central entity, but by a network of nodes (or users). Some nodes can be validators, instead of just ordinary users. Validators compete to add new blocks for monetary advantages. Each network has its own consensus protocol and governance mechanism. We will discuss both in greater depth in the subsequent sections.

In the previous section, we discussed the structural components of the blockchain. In this section we discuss how those structural components are maintained. For instance, we will discuss how transactions are broadcast and gathered in blocks. The following is a general outline of the blockchain work-flow:

1. New transactions are broadcast to all network nodes.
2. Nodes in the network start verifying the broadcast transactions.
3. Validator nodes group the verified transactions into candidate blocks.
4. In each round, a node is selected, *depending on the consensus protocol*, to broadcast its block.
5. Some or all nodes participate in block validation by executing certain functions defined by the consensus protocol.
6. Nodes express their acceptance of the block by including its hash in the next block they create. That is, the verified block is attached to the blockchain, and nodes update their local copy of the blockchain.

Access control schemes determine which nodes can join the network and execute the consensus protocol; an overview of access control schemes can be found in Table 2.1 [7]. In this work, we focus our attention to public, permission-less ledgers.

	Permissioned	**Permission-less**
Public	All nodes can read and submit transactions. Only authorized nodes can validate transactions.	All nodes can read, submit, and validate transactions.
Private	Only authorised nodes can read, submit, and validate transactions.	Not applicable

Table 2.1: Types of distributed ledgers.

2.3 Consensus Protocols

The properties that a ledger consensus protocol must satisfy are outlined in [8]:

1. *Consistency*: parties have the same view of the log of transactions.
2. *Liveness*: transactions are quickly incorporated.

Consensus protocols regulate nodes to extend the longest chain. If a node chooses to deviate from the protocol, by not extending the longest chain in this instance, the node will incur economic losses. This loss could be in terms of time and financial investments spent to find the new block. It thus follows that consensus protocols are incentive compatible [9].

The most prominent consensus protocol is Bitcoin's proof-of-work (PoW) consensus protocol (also known as the Nakamoto protocol) [1]. Recall the fourth step in the blockchain work-flow in the previous section, where a node was somehow selected to broadcast its block. In PoW, the first node that solves a hash puzzle can broadcast its block to the network. The probability that any given node is going to create the next block is equivalent to the fraction of global hash power that it controls. The process of solving the hash puzzle is called mining. Mining is a computation-intensive process, which requires large investments in computational hardware. As such, miners are incentivised by the use of a block reward and transaction fees [6].

An alternative consensus protocol is proof-of-stake (PoS), first introduced by Sunny King and Scott Nadal [10]. Unlike PoW protocols, no excessive computational power is required to participate in the block creation process. Instead, ownership of a currency, the amount of time it has been held or having a deposit in the network allows the nodes to participate in the processes of transaction validation and block creation. Different variations of PoS have different ways of selecting the next block creator. In Nxt [11], the more tokens that are held in an account, the greater the chance that account will earn the right to generate a block. Another variation is Peercoin [10], which uses the idea of 'coin age': older (and larger) sets of coins have a greater probability of signing the next block. In contrast to PoW protocols, PoS protocols introduce predictability: since the stakes are public, each node can predict, with reasonable accuracy, which address will next win the right to create a block.

Of course, regardless of the protocol in place, there may be nodes that deviate from

the protocol to maximise their own payoffs. Game theoretical models have been developed in the literature to analyse the interactions among the nodes and the incentive compatibility of such consensus protocols [9].

2.4 Blockchain Governance

Before discussing governance on the blockchain network, it is important to note that the blockchain network does not exist in a vacuum. Instead, as outlined in [12], the blockchain network is but one layer in the blockchain stack. Each new layer of the stack inherits the protocols and rules of the layer below, including the lower layers' governance. Internet governance, for instance, could have consequences as to who can be allowed to participate in a blockchain network and to what degree. For example, Bitcoin miners in China face the infamous Great Firewall of China, through which all internet packets must pass. This adds technical overhead, which limits the bandwidth and increases the latency of packets transmission [12].

We use the definition of blockchain governance derived in [13]:

'*The means of achieving the direction, control and coordination of stakeholders within the context of a given blockchain project to which they jointly contribute.*'

For the various entities of the blockchain network to coordinate and the network functions without conflicts, a governance mechanism is required. A governing mechanism here is composed of two stages:

1. *Deliberation*. This is where the nodes participating in the blockchain protocol get to decide on a proposal, and whether to adopt it or reject it.
2. *Execution*. This step involves upgrading the blockchain with the adopted proposal or change.

In this work, we will mostly be focusing on deliberation. Unlike centralised forms of governance, participants in *most* blockchain protocols hold the right not to abide by changes to the system. If the majority of the nodes in the protocol vote to upgrade the protocol and move to the upgraded protocol, there exists a risk of a significant portion of the nodes choosing not to upgrade. The opposite scenario also holds, where the majority vote to not upgrade and stay on the original protocol. The risk here is that of a *fork*. Forks can take different forms:

- *Soft forks*. These are *backward-compatible* upgrades, meaning that the upgraded nodes can still communicate with the non-upgraded ones. This is typically in the form of adding a new rule that doesn't clash with the older rules [6].
- *Hard forks*. These are *backward-incompatible* software upgrades, and they occur when nodes add new protocol rules in a way that conflicts with the rules of old protocol. New nodes can only communicate with others that operate the new version only. As a result, the blockchain splits, creating two separate networks: one with the old protocol, and one with the new protocol [6].
- *Velvet forks*. First introduced in [14], velvet forks offer a mechanism that allows

for gradual deployment without harming the miners that haven't upgraded to the new rules. Nodes that upgrade to new rules are still compatible with those that don't.

It is predicted that an absence of an apt governance mechanism can stall and result in a sub-optimised blockchain network [12], leading to frequent soft and hard forks in the network.

Terminology

We first go through some of the terminology that will become relevant throughout the remaining of this work.

- **Utility function**: Given a set of outcomes facing an individual, and over which the individual has a preference ordering, a utility function assigns a real number to each alternative, in such a way that alternative *a* is assigned a number greater than alternative *b* if, and only if, the individual prefers alternative *a* to alternative *b* [15].
- **Rational actors**: An individual is a rational actor if they act in such a way to maximise one's utility; that is, they take the best action available according to their preferences [16].
- **Smart contract**: A smart contract is a computer program or a transaction protocol which is intended to automatically execute, control or document legally relevant events and actions according to the terms of a contract or an agreement. In the context of blockchains, it is deployed on, and ran by, the blockchain network.

2.4.1 Deliberation

In terms of deliberation, the first stage of a blockchain governance mechanism, different approaches could be implemented. These different approaches are on-chain governance (governance by the infrastructure), off-chain governance (governance of the infrastructure), or a mixture of the two [12].

On-chain governance, or governance by the infrastructure, refers to governance by rules hard-coded on the blockchain. It can contain rules that come from within the network community (endogenous) and rules imposed from outside the network community (exogenous) [12]. As an example, for an application deployed on top of a blockchain network, its endogenous rules include the technical rules contained in its smart contracts, and the underlying protocol of the blockchain network would be exogenous. On-chain governance is efficient, transparent, auditable and predictable. Decision-making procedures to reach consensus vary between different blockchain systems. However, a general form of such procedures is that proposals are submitted on-chain, voted on and if agreed upon are deployed on the blockchain test-net for a designated amount of time. If the final vote is in favour after the test-net deployment, then the proposal is incorporated in the main blockchain [17]. Generally, on-chain governance models tend to be fairly decentralised, have quicker turnaround times for

changes and the chances of a hard-fork are reduced. It can, however, suffer from low voter turnout and be manipulated by users with greater stakes [12].

Off-chain governance, or governance of the infrastructure, refers to all forces that subsist outside of a technological platform, but nonetheless influence its development and operations [12]. These rules operate at the social or institutional level, rather than at the technical level. In other words, it resembles the traditional governance mechanisms with relative distribution of power between the different blockchain entities. Off-chain governance rules are implemented via procedures that are not as inflexible as those of a code-base. This allows for better handling of edge cases or unforeseen problems. Descision-making procedures to reach consensus differ among blockchain systems. In Bitcoin, for example, proposals are submitted to the core developers and deliberated through the community before being merged. One of the issues with off-chain governance is lack of incentives for the proposers, which can lead to a small group of developers submitting proposals and hence centralisation [17].

2.4.2 Execution

The process of upgrading the existing blockchain with the proposed changes differs across various blockchain platforms. In blockchains which implement an off-chain governance method, this is generally done via the core developers releasing the updated blockchain software, relying on the miners to update their software and hence the remaining nodes.

In on-chain models of governance, execution generally takes place once the proposal has had a final vote on the blockchain, in which case the software is either updated automatically on the chain (like with self-amending ledgers) or updated manually by the core developers.

Compilers have been proposed, such as in [18], which take a blockchain and turn it into an updatable blockchain.

2.5 Examples of Blockchain Governance

In this section, we go through some examples of decentralised organisations and blockchain platforms, and look at their governance mechanisms and any witnessed failures to date.

To preface, governance failures arise when there are failures to direct, control or coordinate stakeholders in a blockchain system. These problems often result in community splits, or hard forks, which in turn results in fragmented community resources and even a decline in the overall value of the system. We will observe in this section that a community split is likely imminent when the preferences of the different actors in the system have very little to none in common or are completely misaligned.

2.5.1 Bitcoin's Governance

Bitcoin takes an off-chain governance approach, implementing a proof-of-work (PoW) protocol. Proposed protocol updates are usually shared amongst the core developers

in the form of formal improvement proposals, also known as Bitcoin Improvement Proposals (BIPs). The e-proposals are then deliberated through social media and discussion groups. An implementation of the proposal is offered by the individual or the group who proposed the update, or by other core developers. If no implementation is offered, the proposed update runs the risk of abandonment. The proposal is then deliberated through the wider community through various discussion group and other online channels before being merged or deployed by the core developers, and later adopted by miners and other nodes [12].

Bitcoin's governance mechanism is not split-proof, however. If we look closely at the main actors in the Bitcoin network deliberation process (miners, core developers and users), we observe that the different actors have different incentives – and therefore different preferences.

- Miners' incentives are block rewards and transaction fees, so we assume that their most preferred outcome is to maximise their rewards.
- Core developers' incentives are harder to specify; however, we assume that their most preferred outcome is to ensure the maintenance of the security and integrity of the platform.
- Users' incentives include the increase of the value of their holdings and the enhancement of the functionalities of the system, so we assume that their most preferred outcome is to encompass either.

We assume each of those actors to be rational, so that they favour outcomes that maximise their own utility. In 2017, the Bitcoin community attempted to reach consensus to solve an outstanding conflict between miners, who proposed a larger block-size, and core developers, who argued against a larger block-size for the sake of security. Different participants had different visions of which side of the debate maximised their payoffs. Eventually, this lead to the birth of a new chain, Bitcoin Cash, via a hard fork [19].

2.5.2 Ethereum's Governance

Ethereum takes an off-chain governance approach, which implements a PoW protocol (as of the time of writing) [20]. Unlike Bitcoin, the Ethereum network is not only made to support a cryptocurrency (Ether), Ethereum is a smart contract platform which allows entities to leverage blockchain technology to create numerous different digital ledgers and can be used to create additional cryptocurrencies that run on top of its blockchain. Like Bitcoin, proposed protocol updates in Ethereum also take the form of formal improvement proposals, also known as Ethereum Imptorvement Proposals (EIPs), which are backed by substantial technical knowledge. The proposed updates are then deliberated through the community and the core developers get a sense for whether or not miners will agree to upgrade their software before rolling out the update [12].

2.5.3 The DAO's Governance

The DAO was an investment fund, in the form of a smart contract, deployed on the Ethereum blockchain network. The idea was that, after an initial fundraising phase, the general public could make proposals to the DAO community, and the DAO token holders could vote on what projects to fund. Thus, the DAO operated a strict on-chain governance mechanism on an immutable blockchain network. Shortly after the end of fundraising, an attacker exploited a re-entrancy bug to drain funds available to the token holders [21]. Although the attack was spotted immediately, it could not be stopped as it was running on the Ethereum blockchain, which the DAO, as a higher layer, could not influence. The Ethereum community faced a decision: let the attacker keep the funds or modify the Ethereum protocol to alter its state and course of operations (via a hard fork). The main conflict of preferences was manifested as follows [12]:

- The investors' and a part of the community's most preferred outcome was to retrieve their stolen tokens, which required a backward-incompatible update.
- Another part of the community preferred to keep the Ethereum blockchain unaltered for the sake of the platform's integrity.

Each group acted to maximise their own utilities, and not all of the community chose to upgrade to the altered chain, which resulted in a community division [21]. The upgraded chain became known as Ethereum and the original as Ethereum Classic [12].

2.5.4 The Dash Governance System

Dash is a PoW, on-chain governed cryptocurrency that uses a 'budget and masternode voting system' to govern and fund the underlying blockchain's development and maintenance [22]. Each masternode has one (public) vote when deciding which proposals to approve and which to fund. Masternodes must provide a level of service to the network and have a bond of collateral to participate, which is $1,000$ DASH. In each voting cycle (approximately a month), project proposals are submitted and then voted on. The DGS implements an extension of participatory budgeting systems [23]: *Fuzzy Threshold* voting [24], where voters express 'Yes', 'No' or 'Abstain' opinion for each proposal. A proposal is passed if the number of 'Yes' votes, V_{Yes}, minus the number of 'No' votes, V_{No}, is at least 10% of all votes recorded, n. The net total of yes votes must exceed 10% of the total masternode count at the time votes are tallied in order to pass. That is, a proposal is *passed* if:

$$V = (V_{Yes} - V_{No}) \geq 0.1 \times n \tag{2.1}$$

The winning proposals are awarded in the order of the margin by which they are passing, V, until either the entire budget is allocated or no more passing proposals exist. If there are two proposals with the same margin, V, then the one with a larger transaction hash (of the 5 DASH proposal fee) is ranked higher[1]. If a proposal has passed the voting threshold but insufficient funds remain to pay the full amount requested, it will

[1]This can be verified by directly viewing the governance source code of Dash: `https://github.com/dashpay/dash/blob/master/src/governance/governance.cpp`

not receive partial funding. Instead, any smaller proposals which have also passed the threshold that will fit in the budget will be funded, even if they have lower net approval than the larger proposal [22]. If there are more passing proposals than the available block reward can provide for, the proposals with the most yes votes will pass first, creating a cut-off point for less popular proposals.

The governance budget is funded through various channels. When new blocks are mined, 45% of the block reward is reserved for the miner, 10% for the budget and 45% for the masternodes' reward. Submitting proposals also comes at a cost of 5 DASH to ensure that only serious proposals are voted on [25].

2.6 Proposed Governance Models

In this section, we look at governance models that have been proposed to deal with the short-comings of the deployed models or to provide better alternatives. We will look at what short-comings they address and what improvements they offer.

2.6.1 Futarchy Governance

Proposed by economist Robin Hanson [26], the Futarchy model represents the manifesto of 'voting on values and betting on beliefs'. Hanson cites an example of a public company that could hold its chief executive accountable to achieving a particular stock price over a given period of time. This is the chosen metric. Those who believe in the CEO can invest in a 'yes' token, thereby supporting the future success of the company and positioning themselves to get paid out if they are correct. Participants who don't believe in this outcome can invest in a 'no' token, and receive a reward if they are correct.

Hanson argues that traditional voting systems suffer from 'voter apathy', where individuals do not have enough incentive to vote since the probability that their vote will have in effect is insignificant. Futarchy, instead, offers monetary incentives if the individual makes the correct prediction. Another prominent distinction is that, in Futarchy, the participatory governance process encourages focusing more purely on proposals rather than personalities of the leaders or other social affairs. There are, however, many arguments against Futarchy. For example, given that the prediction market is zero-sum and that participation has guaranteed nonzero communication costs, it is therefore not rational to participate. Additionally, it is unclear how consensus could be reached on what metric to use.

Buterin, one of the Ethereum [20] co-founders, outlines, in further detail, the arguments for and against Futarchy, and how it could function in DAOs[2]. Buterin outlines a particular aspect of futarchy that can help counter the *initial* centralisation of a protocol, which is a by-product of the initial one-time token issuance and presale in pursuit of generating *initial* funding and *initial* governance. Buterin argues that if a new pro-

[2]V. Buterin, 'An Introduction to Futarchy', Ethereum blog, 21 August 2014. https://blog.ethereum.org/2014/08/21/introduction-futarchy/ (visited on 10/25/2020)

tocol starts off issuing itself as a futarchy from day one, then that protocol can achieve incentivisation without centralisation.

2.6.2 Quadratic Voting Governance

The motivation for quadratic voting [27] lies in the inability of traditional voting systems in allowing voters to express the intensity of their preference or knowledge towards a certain proposal. The idea is that if an individual voter has a strong preference for, or against, a certain proposal, they could allocate more votes towards it. However, as the number of votes an individual places rises, the monetary value associated with each vote rises quadratically. For example, if a voter places 1 vote they pay 1 coin, if they place 2 votes they pay 4 coins, if they place 3 votes they pay 8 coins, and so on. At the end of each voting cycle, all coins placed are redistributed evenly across the voters. In short, quadratic voting allows voters to show the intensity of their support for a given proposal by placing several votes for it – at the expense of their ability to vote on other issues.

Quadratic voting also makes it difficult for a malicious coalition to take control over the system (by winning many proposals), as it will inevitably take the coalition a large number of cycles and a large amount of coins to be able to do so. Further, quadratic voting is proven to be optimal when there's a need to make a fixed number of collective decisions [27]. However, it is unclear how consensus is reached on the process of selecting a proposal to vote for or against in the first place.

When it comes to blockchain implementations, EximChain [28], which has concluded development in November 2020, implemented a quadratic voting mechanism. The intention behind implementing quadratic voting was that it would decrease the likelihood of anonymous attacks or unforeseen disruptions to the blockchain.

2.6.3 The Treasury Model

A treasury system was introduced in [29] to provide a community controlled, decentralised and collaborative decision-making mechanism for the sustainable funding of the blockchain development and maintenance. The treasury model implements *liquid democracy*, an amalgam of representative and direct democracy. This system of governance is similar to that of Dash, since in each treasury period project proposals are submitted and voted for, with the top-ranked ones granted funding from the treasury. The proposed treasury system is different to that of Dash's in several ways. First, ballots have privacy in the treasury system; votes are not linked to identities. This helps to reduce the chances of voter coercion. Second, each vote's weight is proportional to the corresponding stake. The system utilises the knowledge of community experts in the decision-making process by allowing each voter to either vote directory or delegate their voting power to an expert. Third, the proposed system collects funding via minting new coins and donations in addition to taxes from the miner's block reward.

To participate in the decision-making process, interested stakeholders can register themselves as voters or experts to participate in the decision-making process by locking an amount of their stake in the underlying cryptocurrency. The voter's voting power

is proportional to their locked stake and the expert's voting power is proportional to the amount of voting power delegated to them. The registration process happens in the pre-voting epoch of the treasury cycle, inside which proposals are also submitted (by any member of the community). At the beginning of the voting epoch, there is a voting committee selection stage, during which, a set of voting committee members will be randomly selected from the registered voters who are willing to be considered for selection to the committee. The chances of being selected is proportional to locked stake. The voters and experts can then submit their ballots in the ballot casting stage. For each project, voters can delegate to different experts. At the post-voting epoch, the voting committee members jointly calculate and announce the tally result on the blockchain. Proposals are short-listed if at least 10% of all votes voted 'yes' in their favour. These are ranked according to their score and funded in turns until the treausry fund is exhausted. Finally, in the execution stage, the winning projects are funded, and the voters, experts and voting committee members are rewarded accordingly. These transactions will be jointly signed and executed by the voting committee.

The authors of the proposed model prove that such implementation is resilient up to 50% of malicious participants. Furthermore, they argue that such integration of community-wide knowledge, skills and expertise of members is fundamental for long-term sustainability. The first practical deployment of the treasury model is in Cardano [30].

Chapter 3

Desired Properties of Blockchain Governance

To date, there has been no formal definitions of the desired characteristics of blockchain governance. In this chapter, we utilise *some* of the insights from real-world examples, the proposed models we have covered, social choice theory and game theory, in order to derive a set of desired characteristics for blockchain governance.

Note that we do not claim that a system satisfying the following properties is guaranteed to face no governance problems or that a system not satisfying one of those properties will face governance problems. Instead, the aim here is for those properties to help minimise the chances of governance problems. With each property, except for Non-Optimal Division, we will assess whether the Dash Governance System (DGS) satisfies the property at hand as an evaluation example. The intention here is to demonstrate the ease of determining whether a blockchain governance system satisfies a certain property.

3.1 Privacy

Definition 1. *A governance system meets this criterion when any votes cast are ensured to be private*.

As outlined in the treasury model [29], having the ballots public can have several disadvantages, namely that a malicious group can identify voters who voted against its interests in order to coerce or manipulate them to vote for the malicious group's interests. Note, however, that coercion-resistance should be identified as a separate property on its own (as defined in [31] or in [32]); having private ballots alone does not guarantee coercion-resistance [33]. Another, more subtle, negative repercussion of public voting is that a voter can be ostracised from their community or their group if they vote in such a way to oppose the collective vote of their community or group. Finally, as shown in [24], even a single voter can completely change the final outcome if they know the voting ballots of other voters. Therefore, our first desired characteristic is for ballots to be private.

The process of evaluating whether a blockchain governance system meets this criterion differs depending on whether the underlying network is governed on-chain or off-chain. With off-chain voting, ensuring the ballots are private is similar to ensuring the votes are private in a traditional voting system with anonymous ballots. For evaluating, however, whether a blockchain governance system satisfies this criterion or not, the relevant documentation, source code or referencing work should be examined.

Evaluation example. As reported in [29] and by observing no mechanism in making the votes private in Dash's source code [25], we can clearly identify that votes in Dash are not private. Therefore, the DGS does not satisfy this property.

3.2 Verifiability

Definition 2. *A governance system meets this crietion if voters are able to verify that their votes were legitimately incorporated into the collective vote (individual verifiability)* ***and*** *observers are able to verify that the announced result corresponds to the sum of all votes (universal verifiability).*

Note that individual verifiability can be further broken down into two further properties: classical individual verifiability (a voter can verify that their vote has been counted correctly based on a document representing the received votes, without being able to reconstruct their choice from that document) and constructive individual verifiability (a voter can verify that their vote has been counted correctly by reconstructing their choice from a document representing the received votes), as defined in [34]. In this work, we assume individual verifiability is satisfied if at least one of those two sub-properties is satisfied.

Having those two properties is essential to maintain the participants' confidence in the system as well as to ensure the integrity of the voting system. The challenge of having verifiable votes, whilst also having anonymous ballots, in off-chain governed systems is similar to that faced in modern governmental elections. Systems that overcome this challenge – end-to-end verifiable voting systems – have been proposed in the literature, the most prominent of which is David Chaum's *Secret-Ballot Receipts: True Voter-Verifiable Elections* [35], where visual cryptography techniques are used to allow voters to verify their votes were accurately counted (only individual verifiability is achieved).

In on-chain systems, we assume that both properties are guaranteed in permission-less public ledgers where votes are not private. There are several examples on systems that satisfy such properties. For instance, the blockchain electronic voting system designed in [32] offers both private ballots, individual verifiability and universal verifiability by using cryptographic primitives based on Elliptic-Curve Cryptography (ECC), pairings and Identity Based Encryption (IBE). Similarly, the blockchain voting system designed in [36] offers ballot privacy and coercion-resistance, but *only* universal verifiability.

Evaluation example. We have previously determined that votes are not private on Dash. Since Dash is a public ledger, we therefore, as a consequence, have individual and universal verification satisfied.

3.3 Incentivised Participation

Definition 3. *A governance system meets this criterion if it has at least one clearly defined, direct incentive for voters to participate in the voting process.*

Having incentives for participants to engage in the voting process can help maximise voter participation. Maximising voter participation is important, as research conducted in [37] shows that an engaged community is set to function better than a passive one. An obstacle against maximising voter participation is the issue of rational ignorance [38], where voters refrain from acquiring knowledge when voting, or when delegating their vote, since the cost of acquiring that knowledge exceeds any expected potential benefits. Similarly, it could also be interpreted as refraining from engaging in the voting process all together using the same rationale. Therefore, having participation incentives in place that justify the cost of engagement can lead to higher voter participation.

Evaluation example. One of the incentives present in the DGS is the mining reward. We can see that 45% of the mining reward is kept as a reward for masternodes (nodes with at least 1000 DASH, who get to vote) for their participation in the governance process [22]. This is a direct incentive to participate, and therefore the DGS satisfies this property.

3.4 Pareto Efficiency

This property was chosen as a 'safety' condition to help reduce the chances of a severe misalignment, or conflict, of preferences of the different participating groups in a governance system. This is done by prioritising a set of proposals P over another set of proposals P' if each voter prefers P at least as much as P' and at least one voter strictly prefers P over P' [39]. The property also ensures that a voting system always terminates with a winning proposal or a winning set of proposals as long as at least one voter approved of at least one proposal. By far, this is the most important property as it addresses the fundamental challenge we observed earlier in Section 2.5: the misalignment of preferences.

Background

The input a voter provides into the system can differ depending on what voting system is being used. In the literature, definitions of Pareto efficiency have been devised for *single-winner preferential (or ranked) voting systems*, where the voter's input is an explicit set of ranked preferences, and for *multi-winner approval voting systems with a fixed number of winners*, where the voter's input is a set of alternatives they approve of. For example, the authors in [40] devise a single-winner ranked voting system and define a Pareto efficient system to be a system where 'if there exist two candidates x and y such that no voter prefers candidate y to x, and at least one voter prefers x to y, then the voting system should never declare y the winner'. Another example is how the authors in [39] define Pareto efficiency, for a multi-winner approval voting system (with a fixed number of winners), to be a voting system such that every voter prefers

the winning set at least as much as any other set and at least one voter strictly prefers the winning set over any other set.

The blockchain voting system we have been covering as an evaluation example (Dash) is a variant of *Participatory Budgeting* [23] systems. In particular, the Dash variant has the following properties:

- The voter can give a 'Yes', 'No' or 'Abstain' vote on each proposal as input, instead of only a set of the proposals they approve of (*approval voting*).
- For each project to be considered for selection, it has to meet a certain threshold of approval votes (*threshold approval voting*).
- Each selected project is allocated its entire cost, while each un-selected project is allocated nothing (the algorithm is indivisible or discrete).

That is, the DGS could be expressed, with detail, as an indivisible participatory budgeting system with threshold approval voting. Note that such a system is similar to multi-winner approval voting systems with variable number of winners (with the addition of the budget); and as outlined in [41], the study of such systems has only recently been initiated.

To date, there has been no Pareto efficiency definitions derived for such systems in the literature. We define the necessary preliminaries and derive such a definition, as an extension of the work done in [39] and [24], in this section. Note that we particularly chose Dash as an evaluation example here because a definition that caters for the various parameters that Dash presents will be able to cater for other variants with similar or a smaller number of parameters.

Preliminaries

To start, we define the notation we intend to use. For any given set S, we use:

- 2^S to denote the powerset (the set of all subsets) of S, and
- $(2^S)^n$ to denote a set of n subsets of S.

Then, we fix:

- a finite set of voters, $N = \{1, ..., n\}$
- a finite set of proposals, $P = \{p_1, ..., p_m\}$, where $p_j = (proj_j, b_j)$ with $proj_j$ being the project implemented by proposal p_j and b_j being the requested amount to fund p_j
- a budget, $B \in \mathbb{R}_0^+$, which is used to fund the set of winning proposals

Voters are then asked to choose a set of proposals $X \subseteq P$ of variable size on the basis of a preference profile $\boldsymbol{A} = (A_1, ..., A_n)$ submitted by the voters where each ballot A_i consists of 3 sets $\{A_i^{Yes}, A_i^{No}, A_i^{Abstain}\}$ such that A_i^{Yes} is the set of proposals which voter i gives a 'Yes' vote, A_i^{No} is the set of proposals which voter i gives a 'No' vote and $A_i^{Abstain}$ is the set of proposals which voter i gives an 'Abstain' vote.

The choice of such X can be delegated to a voting rule, F, which is a function mapping a pair of any given profile of approval ballots and a budget to a set of committees (where a committee is a set of proposals). We describe such F as follows:

$$F : (2^P \times 2^P \times 2^P)^n \times \mathbb{R}_0^+ \to 2^{2^P} \tag{3.1}$$

Note that unlike the modelling in [39], the output of F could be a set containing only the empty set (e.g. if there are insufficient funds to fund any of the proposals in P). This is to ensure that our model is applicable to as many voting rules as possible. The *exact* definition of F is specific to each voting system. For example, depending on the specific choice of F, F can be *resolute*: $|F(\boldsymbol{A})| = 1$ for every profile $\boldsymbol{A} \in (2^P \times 2^P \times 2^P)^n$. Likewise, F can be *irresolute* if more than one winning committee is possible. In this case, Dash is a *resolute* system.

In this model, voters express their views by specifying which proposals they approve of (not which committees they prefer). To refer to the preferences of the voters, we will first need to make a few assumptions.

- First, assume that every voter $i \in N$ truthfully holds the sets $A_i^{Yes}, A_i^{No}, A_i^{Abstain} \in P$.
- Second, each voter i has a unique **weak** preference order, $\succcurlyeq_{A_i^{Yes}}$, on 2^P satisfying the following condition for all committees $X, X' \in 2^P$:
$$X \succcurlyeq_{A_i^{Yes}} X' \Leftrightarrow |A_i^{Yes} \cap X| \geq |A_i^{Yes} \cap X'|$$
- Third, each voter i has a unique **strict** preference order, $\succ_{A_i^{Yes}}$, on 2^P satisfying the following condition for all committees $X, X' \in 2^P$:
$$X \succ_{A_i^{Yes}} X' \Leftrightarrow |A_i^{Yes} \cap X| > |A_i^{Yes} \cap X'|$$

In other words, we say that voter i prefers X at least as much as X' (or weakly prefers X to X') if $|A_i^{Yes} \cap X| \geq |A_i^{Yes} \cap X'|$ and strictly prefers X to X' if $|A_i^{Yes} \cap X| > |A_i^{Yes} \cap X'|$.

Given that the outcome of F is a set of committees, we need to extend the preference orders defined so far to cater for the outcomes of F. Like in [39], we will use the Kelly extension [42] to achieve this:

- Each voter i has a unique **weak** preference order, $\succcurlyeq^{EXT}_{A_i^{Yes}}$, over a pair of sets of committees, $\mathcal{X}, \mathcal{X}' \in 2^{2^P}$, satisfying the following condition:
$$\mathcal{X} \succcurlyeq^{EXT}_{A_i^{Yes}} \mathcal{X}' \Leftrightarrow (\forall X \in \mathcal{X})(\forall X' \in \mathcal{X}')X \succcurlyeq_{A_i^{Yes}} X'$$
- Each voter i has a unique **strict** preference order, $\succ^{EXT}_{A_i^{Yes}}$, over a pair of sets of committees, $\mathcal{X}, \mathcal{X}' \in 2^{2^P}$, satisfying the following condition:
$$\mathcal{X} \succ^{EXT}_{A_i^{Yes}} \mathcal{X}' \Leftrightarrow (\forall X \in \mathcal{X})(\forall X' \in \mathcal{X}')X \succ_{A_i^{Yes}} X'$$

Note that, unlike $\succcurlyeq_{A_i^{Yes}}$, $\succcurlyeq^{EXT}_{A_i^{Yes}}$ is not a complete relation: not all pairs of sets of committees will be ranked.

Definitions

With all the preliminaries defined, it is now appropriate to devise a Pareto efficiency definition for *participatory budgeting systems with approval voting*.

Definition 4.1. F is *Pareto efficient* if for any profile $\boldsymbol{A}$ and any two committees $X, X' \in 2^P$ with $X \succcurlyeq_{A_i^{Yes}} X'$ for all voters $i \in N$ and $X \succ_{A_i^{Yes}} X'$ for at least one voter $i \in N$, it is the case that $X' \notin F(\boldsymbol{A})$.

In other words, a Pareto efficient voting rule should not have a committee X' amongst its winning committees *when* there is another committee X that is weakly preferred by all and strictly preferred by at least one of the voters to X'. We limit the definition here to committees for easier evaluation.

Definition 4.2. A participatory budgeting voting system is *Pareto efficient* if its voting rule is *Pareto Efficient* as per Definition 4.1.

Observe that this definition of Pareto efficiency has the following consequences:

- If the winning committee is $X' = \emptyset$ and at least one voter approved of at least one proposal, then the underlying voting rule violates Definition 4.1.
- If the winning committee is $X' = \emptyset$ and no voter approved of any proposal, then the underlying voting rule satisfies Definition 4.1.

This is to say that if at least one voter approved of at least one proposal, the Pareto efficiency property guarantees a non-empty outcome. And, if no voter approved of any proposal, then the property guarantees that things remain as they are.

Evaluation example

We now examine if the Dash Governance System (DGS) meets the derived definition of this property. We model the DGS voting rule on the information outlined in Section 2.5.4. The DGS voting rule, F_{DGS}, is a *resolute* voting rule that returns a set of winning proposals according to the following rules.

1. We fix a finite set of masternodes (voters), $N = \{1, ...n\}$, a finite set of proposals, $P = \{p_1, ..., p_m\}$, and a budget $B \in \mathbb{R}_0^+$ as before.
2. Let $score(p_j) = \sum_{i=1}^{n} pref(p_j, i)$ be a function such that given a certain proposal $p_j \in P$, it returns a score for this proposal, where $pref(p_j, i)$ returns 1, -1 or 0 if voter i votes 'Yes', 'No' or 'Abstain' respectively.
3. Let $P_{passed} = \{p_{x_1}, ..., p_{x_z}\} \subseteq P$ be a sorted list of proposals such that $score(p_{x_1}) \geq ... \geq score(p_{x_z})$, and for each $p_j \in P_{passed}$ it holds that $score(p_j) \geq 0.1 \times n$.
4. A sorted list of winners $W = \{p_{y_1}, ..., p_{y_z}\}$ consists of the proposals from P_{passed} where $score(p_{y_1}) \geq ... \geq score(p_{y_z})$ such that $\sum_{w=1}^{z} b_{y_w} \leq B$ (where b_{y_w} is the amount requested to fund p_{y_w}).

We now prove, by giving a set of counter examples, that the DGS does not meet the derived Pareto efficiency property.

Proposition. Fix a set of voters $N = \{1, ...n\}$, a set of proposals $P = \{(proj_1, b_1), (proj_2, b_2), ..., (proj_m, b_m)\}$ and a budget B. The DGS will fail to meet the derived definition of Pareto efficiency whenever:

- $score(p_{x_1}) \geq ... \geq score(p_{x_{z-2}}) \geq score(p_{x_{z-1}}) > score(p_{x_z})$ such that proposals $p_{x_1}, ..., p_{x_{z-2}}, p_{x_{z-1}}, p_{x_z} \in P_{passed}$;
- $b_{x_1} + ... + b_{x_{z-2}} + b_{x_{z-1}} > B$ and $b_{x_1} + ... + b_{x_{z-2}} + b_{x_z} \leq B$; and therefore,
- the winning set is $W = \{p_{x_1}, ..., p_{x_{z-2}}, p_{x_z}\}$ instead of $W' = \{p_{x_1}, ..., p_{x_{z-2}}, p_{x_{z-1}}\}$

Proof. Since $score(p_{x_1}) \geq ... \geq score(p_{x_{z-2}}) \geq score(p_{x_{z-1}}) > score(p_{x_z})$, by definition it implies that for each voter i: $|A_i^{Yes} \cap W'| \geq |A_i^{Yes} \cap W|$ and that for at least one voter i': $|A_{i'}^{Yes} \cap W'| > |A_{i'}^{Yes} \cap W|$ (since $score(p_{x_{z-1}}) > score(p_{x_z})$). And, since $W \in F_{DGS}(\boldsymbol{A})$, Definition 4.1 will not be met, and hence the DGS will not meet Definition 4.2.

An instance of such cases where the DGS fail to be Pareto efficient is as follows: fix a set of voters $N = \{1, 2, 3\}$, a set of proposals $P = (proj_1, 3), (proj_2, 4), (proj_3, 5)$ and a budget $B = 7$. Let Table 3.1 express $\boldsymbol{A}$.

i	$pref(p_1, i)$	$pref(p_2, i)$	$pref(p_3, i)$
1	1	-1	1
2	1	1	0
3	1	1	1

Table 3.1: Example preference profile.

This gives us the following scores: $score(p_1) = 3$, $score(p_2) = 1$ and $score(p_3) = 2$. Therefore, $P_{passed} = \{p_1, p_3, p_2\}$. We can see that the only possible winning list is $W = \{p_1, p_2\}$ (instead of $\{p_1, p_3\}$). This is because allocating the amount requested for p_1 ($b_1 = 3$) leaves us with a remaining budget of 4, which cannot fund p_3 (since $b_3 = 5$) but can fund p_2 (since $b_2 = 4$).

We now treat W as a set instead of a list, and let W' be another set such that $W' = \{p_1, p_2, p_3\}$. Given that $F_{DGS}(\boldsymbol{A}) = \{W\}$. From the given table, we can derive the following sets: $A_1^{Yes} = \{p_1, p_3\}$, $A_2^{Yes} = \{p_1, p_2\}$ and $A_3^{Yes} = \{p_1, p_2, p_3\}$. We can immediately observe that $|A_i^{Yes} \cap W'| \geq |A_i^{Yes} \cap W|$ for every voter i and that for voter 1 it is the case that $|A_1^{Yes} \cap W'| > |A_1^{Yes} \cap W|$. However, $W \in F_{DGS}(\boldsymbol{A})$. Therefore, $F_{DGS}(\boldsymbol{A})$ does not meet Definition 4.1. Since the voting rule of the DGS, $F_{DGS}(\boldsymbol{A})$, does not meet Definition 4.1, the DGS does not satisfy Definition 4.2. Thus, the DGS does **not** satisfy our derived criterion of Pareto efficiency.

3.5 Non-Optimal Division

Definition 5. *A self-amending governance system meets this criterion if the conditions, that make it feasible for community division to be a Nash equilibrium, exist.*

For the remaining of this section, we will further dissect and clarify Definition 5. In Section 5.5, we will evaluate Tezos against the definition.

Adapting the work done in [17] to strictly self-amending distributed ledgers, we model such governance systems into a two-player strategy game with the voters, V, and the broader community, C, as the players.

We define the individual actions of each voter in V as:

1. Promote the proposal p.
2. Reject the proposal p, resulting in no change to the ledger's protocol.

Recall that in self-amending blockchains, unlike other on-chain governed blockchains, a participant cannot choose not to upgrade to the protocol of the newly promoted proposal. For instance, if a participant or a stakeholder is firmly opposed to a newly promoted proposal, they cannot choose to stay on the original chain and not upgrade. Instead, they are faced with two choices: stay as a stakeholder or leave the platform (by means of exchanging or selling their stake). We generalise this phenomenon to the broader community of stakeholders, and define the individual actions of each stakeholder in C as:

1. Stay as a stakeholder.
2. Leave the blockchain platform.

We define β as the proportion of voters, V, that voted to promote the proposal p, and γ as the proportion of all stakeholders, C, that leave the blockchain platform in response to the outcome of the voting process. We can now simplify the governance process to a two-player strategy game, using the payoff matrix in Table 3.2 for evaluation, which enumerates the strategies of V and C.

		The Broader Community (C)	
		Leave	**Stay**
The Voters (V)	**Promote**	$\beta S(V)$, $\gamma S(C)$	$\beta S(V)$, $(1-\gamma)S(C)$
	Reject	$(1-\beta)S(V)$, $\gamma S(C)$	$(1-\beta)S(V)$, $(1-\gamma)S(C)$

Table 3.2: The payoff matrix for the blockchain governance game.

The payoff function in Table 3.2, $S(\cdot)$, is the sum of individual payoffs as a result of the decision and it implies the perceived long-term benefit in staying with, or leaving, the given blockchain platform. In choosing to stay, the perceived benefit could be in higher

value for the cryptocurrency, implying more monetary incentive for validating blocks or more competitive transaction fees. In choosing to leave, the perceived benefit could be in avoiding a lower value for the cryptocurrency or finding a better use of their stake elsewhere. For simplicity, we assume no distinction between the value of payoffs of the entities in C and that of the entities in V:

$$S(V) = \sum_{i=1}^{n} S(V_i) = nS_r \tag{3.2}$$

$$S(C) = \sum_{i=1}^{k} S(C_i) = nC_k \tag{3.3}$$

Let the total payoff for each player be 1 such that $S(V) = 1$ and $S(C) = 1$. Observe that there are four possible scenarios when proposal p is subject to a vote:

1. p is promoted:
   (a) The majority of C stays, or
   (b) The majority of C leaves.
2. p is rejected:
   (a) The majority of C stays, or
   (b) The majority of C leaves.

In this work, we will focus on scenarios 1(b) and 2(b) as we are interested in scenarios of community division, which correspond to the top-left and bottom-left cells, respectively, of the payoff matrix in Table 3.2.

Under certain conditions, it is *possible* for β and γ to have certain values such that either scenario 1(b) or scenario 2(b) is a Nash equilibrium in the payoff matrix. The optimal outcome, then, would be for the majority of C to leave the given blockchain platform.

To clarify, an example of such conditions is systems that equate the weight of a vote to its associated stake, when a minority of C holds most of the stake, and in turn holds most of the votes of V. In this case, it is *possible* for this minority to promote p whilst the majority of C rejects p and leaves in response (scenario 1(b)). Therefore, in this case, there is a condition in place that makes such a Nash equilibrium *feasible* and such a governance system would not meet Definition 5 as a result.

Chapter 4

Overview of Tezos

Tezos [2, 43] is a *self-amending* distributed ledger that implements an (optional) delegated proof-of-stake protocol (DPoS), which is similar to liquid democracy where owners of the underlying currency can choose to delegate their stake as part of the governance process. In this chapter, we provide an overview of the current Tezos consensus protocol (the *Edo* protocol) and governance process.

4.1 DPoS in Tezos

In Tezos, a participant in the consensus protocol needs to have a minimum stake of $8,000$ tokens, which is called a *roll*. If a participant does not have enough stake to participate on their own or does not want to set up the needed infrastructure, they can use delegation. Therefore, in Tezos, participants in the consensus algorithm are called delegates. At each block-height, a number of delegates are randomly selected to become *bakers* and *endorsers*. Bakers and endorsers have different roles [44]:

- **Baker**. A baker collects transactions gossiped over the peer-to-peer network, which they then assemble into a block.
- **Endorser**. An endorser signs the best block (according to a certain criteria) they have heard of at a given block-height.

Baking rights and endorsing rights are determined at the beginning of a cycle (a group of 4096 blocks) by a *follow-the-satoshi* strategy starting from a random seed computed from information already found on the blockchain. Delegates are considered *active* when they participate in the creation and validation of blocks (and *passive* otherwise). A delegate becomes passive for a cycle when they fail to create any blocks or endorsements in the past 5 cycles. Only active delegates can be selected to be endorsers or bakers.

As an incentive to remain active, delegates are rewarded for their baking and endorsing. Each delegate key has an associated security deposit account. When a delegate bakes or endorses a block, the security deposit is automatically moved to a deposit account. As a counter-measure against double-baking or double-endorsement, the security deposit

is frozen from the delegate's account. The deposit is either released after a number of cycles or burnt in case of proven malicious behaviour.

Each delegate is associated with a set of rolls. Active delegates participate in a lottery to *bake* or *endorse* a block at every block-height in the chain. At each block-height, 32 active rolls are randomly selected and the owners of those rolls make up the list of endorsers for the current cycle. All 32 endorsers have the same weight, with no particular endorser having a priority over the other. Additionally, each height is associated with a *priority list* of delegates, with priority 0 being the highest. This list is obtained by randomly selecting the owner of an active roll for each position in the list. The first baker in the list is the first one who can bake a block at that level. If a delegate is for some reason unable to bake, the next delegate in the list can step up and bake the block. As the draw is independent for each list position, it is possible that the same public key appears multiple times in this list. Participants that do not hold enough tokens or who do not wish to bake blocks can delegate their tokens to another baker. They keep the ownership of their tokens but increase the stake of their delegate in the random assignment of baking slots [45].

A block in Tezos is *valid* when a certain amount of time (a delay) has elapsed between it and its predecessor. This delay, $D(p, e)$, is a function of the priority of the baker, p, and the number of endorsements that the block includes, e:

$$D(p, e) = 60 + (40 \cdot p) + (8 \cdot max(0, 24 - e)) \tag{4.1}$$

That is, the higher the priority and the fewer endorsements a block carries the longer it takes before it can be considered valid. However, if the number of endorsements is greater than 24 then there will be no time penalty [43].

When baking, the baker has to choose which chain of blocks (or branch) they will extend. This is also known as the *fork-choice rule*. In the current protocol, each chain has a 'fitness' score which measures the length of the chain and the 'fitness' of a block is 1 plus the fitness of the previous block. Thus, the best fork choice is the longest chain, which has the highest fitness score [43]. Although the current rule makes evaluation of branches easier and alleviates a baker's uncertainty as to when to publish blocks in order to avoid missing out on late endorsements, work done in [45] and [46] outlines the vulnerabilities of such a rule.

Bakers and endorsers are awarded differently for their participation. A baker's reward for a block, $R_B(p, e)$, depends on the number of endorsements of the block and the priority of the baker, whilst an endorser's reward, $R_E(p_i)$, depends on the priority of the endorsed block's baker, p_i [43]:

$$R_B(p, e) = \begin{cases} 1.25e & \text{if } p = 0 \\ 0.1875e & \text{if } p \geq 1 \end{cases} \tag{4.2}$$

$$R_E(p_i) = \begin{cases} 1.25 & \text{if } p = 0 \\ 0.8333333 & \text{if } p \geq 1 \end{cases} \tag{4.3}$$

That is, the rewards for endorsing or baking a block without the highest priority are significantly smaller than the rewards for baking or endorsing a block with priority 0.

4.2 Governance in Tezos

Tezos's standout feature is its ability to amend itself (e.g. to amend the consensus protocol). The ledger's self-amending nature is of upmost importance to its governance, since delegates, via a voting procedure, can propose, select and test a candidate protocol before activating it. Delegates take part in the amendment procedure with an influence proportional to their stake (i.e. *one roll equates to one vote*) [43]. In this section, we cover both the deliberation and execution aspects of governance in Tezos.

4.2.1 Deliberation

In order to amend itself, Tezos uses an on-chain voting system where delegates participate to propose, select, adopt or reject new amendments. The voting process is currently divided in five governance periods, each period spanning roughly two weeks or 20480 blocks (i.e. 5 cycles). Note that just like block baking and endorsing, participants that do not hold enough tokens or do not wish to participate can also delegate their tokens to a delegate in the voting process. The stake of each delegate is then computed at the beginning of each governance period and stored in a list called the *voting listings*. The following is a breakdown of the five governance periods [43]:

1. *Proposal period.* Delegates can submit protocol amendment proposals using the proposals operation as long as the underlying codebase compiles with the change. Delegates then upvote their preferred proposal or proposals. At the end of a proposal period, the proposal with the most upvotes is selected and a testing-vote period starts. If there are no proposals, no proposals with upvotes of at least 5% of the possible votes, or a tie between proposals, a new proposal period starts. Each delegate can submit a maximum of 20 proposals, including duplicates.

2. *Testing-vote period.* Delegates can cast one vote to test or not the winning proposal using the ballot operation. At the end of a testing-vote period if the *participation* reaches the *quorum* and the proposal has a *super-majority* in favour, a testing period starts. Otherwise it goes back to a proposal period.

3. *Testing period.* A test chain is forked for the entire testing period to ensure a correct migration of the context.

4. *Promotion-vote period.* Delegates can cast one vote to promote or not the tested proposal using the ballot operation. At the end of a promotion-vote period, if the *participation* reaches the *quorum* and the proposal has a *super-majority* in favour, an adoption period starts. Otherwise it goes back to a proposal period.

5. *Adoption period.* The adoption period serves as a buffer time for users to update their infrastructure to the new protocol. At the end of this period, the proposal is activated as the new protocol and a new proposal period starts.

In periods 2 and 4, a delegate can cast a single 'Yea', 'Nay' or 'Pass' vote. A vote is then successful if it has *super-majority* and the *participation* reaches the current *quorum*:

- **Super-majority**. A super-majority is when the number of 'Yea' votes, V_{Yea}, is more than 80% of the number of 'Yea' votes, V_{Yes}, and 'Nay' votes, V_{No}:

$$V_{Yes} > 0.8 \cdot (V_{Yes} + V_{No}) \tag{4.4}$$

- **Participation**. The participation is the ratio of all received votes ('Yea', 'Nay' and 'Pass' votes) with respect to the number of possible votes.

- **Quorum**. The quorum, q, is a threshold for participation, c. In the genesis block, the initial value of the quorum was 80%: $q_0 = 0.8$. After each vote it is updated as follows:

$$q_{i+1} = 0.8 \cdot q_i + c \cdot 0.2 \tag{4.5}$$

  When updating the quorum according to Equation 4.5, the quorum can reach very high values which would make passing new proposals very difficult even if there is large acceptance. As a consequence, in the Babylon update [43], quorum caps were introduced to have a minimum cap of $q_{min} = 0.2$ and a maximum cap of $q_{max} = 0.7$.

4.2.2 Execution

As our focus in this work is on the deliberation stages of governance, we will keep the discussion on the execution stages to a minimum. To understand this stage of Tezos governance, we should first understand the underlying architecture.

The architecture of the Tezos node can be viewed as made up of two parts [43]:

- The *protocol*: it is responsible for interpreting the transactions and other administrative operations. It also has the responsibility to detect erroneous blocks and it always sees only one block chain.

- The *shell*: it has the responsibility of selecting and downloading alternative chains, input them to the protocol, which in turn checks them for errors, and gives them an absolute score. The shell then simply selects the valid head of the highest absolute score. This part of the shell is called the validator. Additionally, the shell includes the peer-to-peer layer, the disk storage of blocks, the operations to allow the node to transmit the chain data to new nodes and the versioned state of the ledger.

When a node launches for the first time, it starts with the genesis protocol and then goes through all previous protocols until it finally switches to the current protocol and the current context. The *context* of the blockchain is the full state of the blockchain shared by all participants or peers. A new block is created roughly every minute, and when the shell receives a new block on the peer-to-peer network, it applies each operation in the block to its current context to compute a new context [2]. This process constitutes the basis of protocol amendment.

With the background of the Tezos node architecture outlined, we can describe, in a high-level fashion, the process of activating or upgrading to a new protocol. Recall the 5th governance period, the adoption period, where at the end of it the chosen proposal is activated as the new protocol. The upgrade procedure can be outlined as follows:

1. In the 5th period, participants update their node software to a version that contains the new protocol before its activation.
2. When the 5th period ends:
    - In nodes that have updated, the new protocol is automatically activated by having the shell expose certain procedural functions to the protocol and applying them as soon as the 5th period ends. These functions change the node's context by changing the protocol when the 5th period ends.
    - In nodes in which the node admin did not update, what happens is as follows:
        - First, note that the procedure of voting and governance periods for a new protocol is embedded in the current protocol. This embedded voting procedure establishes when the vote starts, ends and when the new protocol is finally activated (if the vote was successful). Consequently, the shell is aware of those procedures inside the protocol.
        - Second, since the shell is protocol-independent, as soon as a new block is received by the node that is baked by a baker running the new protocol the shell will automatically start using the protocol associated to this new block.

In short, what gives Tezos its self-amending nature is that all nodes, by design, agree that for a block to be considered valid after the activation of the new protocol, it must follow the rules of the new protocol [43]. That is, no node *can choose* not to upgrade.

Chapter 5

Analysis of Tezos

In this chapter, we will analyse Tezos's governance mechanism with respect to the five desired properties of blockchain governance we derived in Chapter 3.

5.1 Privacy

Recall the definition of the Privacy property in Section 3.1. Ballots in Tezos have the following form [43]:

```
Ballot : {
  source: Signature.Public_key_hash.t ;
  period: Voting_period_repr.t ;
  proposal: Protocol_hash.t ;
  ballot: Vote_repr.ballot ;
}
```

The `Ballot` attributes can be described as follows:

- `source`: the public key hash of the delegate.
- `period`: the unique identifier of each voting period.
- `proposal`: the currently selected proposal.
- `ballot`: the delegate's vote, which can be `Yea`, `Nay` or `Pass`.

Because of the nature of the governance process, specifically the delegation mechanism, ballots have to be public: the public key of the delegate is recorded on the ballot. This is to ensure that when participants delegate their stake to a particular delegate, observers can see how this delegate voted. Therefore, it is clear that ballots or votes in Tezos are not private, and hence it does **not** meet the privacy property [43].

5.2 Verifiability

Recall that, as we defined in Section 3.2, to satisfy this property the governance system in question must satisfy both individual and universal verifiability.

In the previous section, we have shown that votes cast in Tezos are public. This implies that delegates who vote are able to verify that their votes were incorporated into the collective vote, which satisfies individual verifiability. Similarly, any observer can verify that the final result of a vote is indeed the result of the individually cast votes. Therefore, Tezos **does** meet the verifiability property [43].

5.3 Incentivised Participation

Recall that, as we defined in Section 3.3, to satisfy this property the governance system in question must have at least one clearly defined incentive for voters (delegates, in the case of Tezos) to participate in the voting process.

In the previous chapter, we have shown that there are rewards for bakers and endorsers for their services in the consensus protocol. There are certain *indirect* incentives for delegates to participate in the governance process:

- Token owners who delegate are likely to move their delegations if their delegates do not vote as they intend or don't vote at all.
- Delegates have their tokens locked by the protocol. This represents something of value to them at stake. Thus, we assume they want these tokens to remain valuable and therefore this incentivises them to participate in the governance process.

No additional or direct rewards, however, are given to delegates for participating in the governance process (for voting or otherwise). Therefore, to be specific to the definition, there are no clearly defined *direct* incentives in Tezos for voters to participate in the voting process. Hence, Tezos does **not** meet the incentivised participation property.

5.4 Pareto Efficiency

Unlike the evaluation example of Dash in Section 3.4, the Tezos voting process is a multi-stage one:

- Delegates upvote the proposals they approve of in the first governance period (the proposal period).
- Delegates vote whether or not to test the most upvoted proposal in the second governance period (the testing-vote period).
- Finally, delegates vote whether or not to promote the tested proposal in the fourth governance period (the promotion-vote period).

Since the definition for Pareto efficiency defined in Section 3.4 does not cater for multi-stage voting inputs, we will make a one-stage voting process approximation of the

Tezos voting process by outlining the notation we will use, making a few assumptions, and defining the necessary preliminaries.

Notation

The notation we will use is as follows:

- Since each roll corresponds to a single vote, let us refer to voters as rolls, such that E is the set of rolls that are eligible to vote and N is the subset of eligible rolls that do vote.
- Let us denote the first governance period (the proposal period) by I since it is the first stage of voting, the second governance period (the testing-vote period) by II since it is the second stage of voting, and the fourth governance period (the promotion-vote period) by III since it is the third stage of voting.
- In stage I, each delegate can upvote (or approve) a subset of the given proposals. Therefore, we will refer to the set of proposals that delegate i approves of in stage I as $A_i^{Yes,I}$. Only one proposal, the most approved one, will pass to the subsequent stages.
- In stage II, each delegate can vote with 'Yea', 'Nay' or 'Abstain' on whether or not to test the most upvoted proposal. Therefore, we will refer to the set of proposals that delegate i approves of in stage II as $A_i^{Yes,II}$, the set of proposals that delegate i disapproves of as $A_i^{No,II}$ and the set of proposals that delegate i abstains on as $A_i^{Abstain,II}$ such that $|A_i^{Yes,II}| + |A_i^{No,II}| + |A_i^{Abstain,II}| = 1$.
- In stage III, each delegate can vote with 'Yea', 'Nay' or 'Abstain' on whether or not to promote the proposal that passed stage II. Therefore, we will refer to the set of proposals that delegate i approves of in stage III as $A_i^{Yes,III}$, the set of proposals that delegate i disapproves of as $A_i^{No,III}$ and the set of proposals that delegate i abstains on as $A_i^{Abstain,III}$ such that $|A_i^{Yes,III}| + |A_i^{No,III}| + |A_i^{Abstain,III}| = 1$.

Assumptions

Having outlined the notation we will use, it is now appropriate to make the assumptions that will allow us to approximate the Tezos voting process as a one-stage process:

1. First, we will assume that the same set of rolls, N, will participate in each of the three stages, where $N \subseteq E$ and E is the set of all the rolls that are eligible to vote.
2. Second, we will assume that for each roll $i \in E$: $A_i^{Yes,II} = A_i^{Yes,III}$, $A_i^{No,II} = A_i^{No,III}$ and $A_i^{Abstain,II} = A_i^{Abstain,III}$. This simply means that if a delegate is voting with 'Yea', 'Nay' or 'Abstain' on the given proposal in stage II, they would also vote on it with 'Yea', 'Nay' or 'Abstain' (respectively) in stage III.
3. Third, we will assume that if $p \in A_i^{Yes,II}$ (or $p \in A_i^{Yes,III}$, since $A_i^{Yes,II} = A_i^{Yes,III}$) for some proposal p, then it is also the case that $p \in A_i^{Yes,I}$.

Note that in Tezos $A_i^{Yes,II} \cap A_i^{No,II} \cap A_i^{Abstain,II} = \emptyset$ and $A_i^{Yes,III} \cap A_i^{No,III} \cap A_i^{Abstain,III} =$

$\emptyset$. Thus, it follows from assumptions 2 and 3 that $A_i^{Yes,I} \cap A_i^{No,II} \cap A_i^{Abstain,II} = \emptyset$ and $A_i^{Yes,I} \cap A_i^{No,III} \cap A_i^{Abstain,III} = \emptyset$.

Preliminaries

Given those assumptions, it is now appropriate to approximate the voting process to a one-stage process with the *i*th roll's input in the voting process to be composed of three sets A_i^{Yes}, A_i^{No} and $A_i^{Abstain}$. We define those sets as follows:

- Let $A_i^{Yes} = A_i^{Yes,I}$
- Let $A_i^{No} = A_i^{No,II} = A_i^{No,III}$
- Let $A_i^{Abstain} = A_i^{Abstain,II} = A_i^{Abstain,III}$

We will consider the input of the non-participating, yet eligible to vote rolls as three empty sets or $(A_i^{Yes}, A_i^{No}, A_i^{Abstain})$ where $A_i^{Yes} \cup A_i^{No} \cup A_i^{Abstain} = \emptyset$. Having defined the input for each eligible-to-vote roll, we are now ready to define the Tezos voting rule, F_{Tezos}. Fix a finite set of rolls that are eligible to vote, $E = \{1, ..., e\}$, a finite set of proposals, $P = \{p_1, ..., p_m\}$ where $p_j = (proj_j, 0)$, and a budget of $B = 0$:

$$F_{Tezos} : (2^P \times 2^P \times 2^P)^e \times 0 \to 2^{2^P} \tag{5.1}$$

Note that the budget and the amount required by each project are all zero since listed proposals are ready to be compiled and do not need financing to be built from scratch. We also note that the output will be a set of a single committee (F_{Tezos} is resolute) that contains either a single proposal or no proposals. That is, a winning committee X in Tezos will have a cardinality of $|X| \leq 1$. The output of the voting rule function follows the same rules and conditions we outlined in Section 4.2.1. That is, for a proposal $p \subseteq P$ to be the winning proposal, the following four conditions need to be met:

- The participation ratio, $\frac{|N|}{|E|}$, must be greater than or equal to the current quorum, q_c. That is,

$$\frac{|N|}{|E|} \geq q_c \tag{5.2}$$

  where $0.2 \leq q_c \leq 0.7$.

- Proposal p must have at least 5% of all possible upvotes. That is,

$$\frac{\sum_{i=1}^{e} g(A_i^{Yes}, p)}{|E|} \geq 0.05 \tag{5.3}$$

  where,

$$g(S, x) = \begin{cases} 1 & \text{if } x \subseteq S \\ 0 & \text{if } x \not\subseteq S \end{cases} \tag{5.4}$$

- Proposal p is the most upvoted proposal. That is, for any other proposal $p' \subseteq P$, it is the case that:

$$\sum_{i=1}^{e} g(A_i^{Yes}, p) > \sum_{i=1}^{e} g(A_i^{Yes}, p') \tag{5.5}$$

- Proposal p has a super-majority. That is, $V_{Yes} > 0.8 \cdot (V_{Yes} + V_{No})$ (Equation 4.4) where,

$$V_{Yes} = \sum_{i=1}^{e} g(A_i^{Yes}, p) \tag{5.6}$$

and

$$V_{No} = \sum_{i=1}^{e} g(A_i^{No}, p) \tag{5.7}$$

Evaluation

We now prove that the Tezos governance system does not meet the derived Pareto efficiency definition.

Proposition. Fix a set of rolls that are eligible to vote $E = \{1, ...e\}$, a subset of such rolls that do vote, $N = \{1, ...n\}$, a set of proposals $P = \{p_1, ..., p_m\}$ where $p_j = (proj_j, 0)$ and a budget $B = 0$. The Tezos governance system does **not** satisfy the derived definition of Pareto efficiency by having, in certain cases, a winning committee $X' \in 2^P$ whilst there exists another committee $X \in 2^P$ such that $X \succcurlyeq_{A_i^{Yes}} X'$ for all rolls $i \in E$ and $X \succ_{A_i^{Yes}} X'$ for at least one roll $i \in E$.

Proof. Recall from Definition 4.1 in Section 3.4, that a voting rule is Pareto efficient if it cannot have committee $X' \in 2^P$ amongst its winning committees *when* there is another committee $X \in 2^P$ that is weakly preferred by all rolls $i \in E$ and strictly preferred by at least one of the rolls $i \in E$. Consider the case when $F_{Tezos}(\boldsymbol{A}) = \{X'\}$, where $X' = \emptyset$. Since $X' = \emptyset$, then at least one of the following statements must be true: (a) the participation ratio has not reached the current quorum, (b) no proposal $p \in P$ had at least 5% of the upvotes, (c) no proposal $p \in P$ had more upvotes than any other proposal, or (d) no proposal $p \in P$ had a super-majority.

1. In the case of $|N| = 0$:
    - Since the participation ratio is zero, then for every other set $X \in 2^P$ there will be no roll $i \in E$ such that $|A_i^{Yes} \cap X| > |A_i^{Yes} \cap X'|$, and hence in this particular edge case Definition 4.1 will be met.
2. In the case of $N > 0$:
    - For (a), (b), (c) or (d), it is sufficient to pick any $X = \{p\} \in 2^P$ such that for at least one roll $i \in E$ we have $p \in A_i^{Yes}$. It then follows that for every roll $j \in E$ we have $|A_j^{Yes} \cap X| \geq |A_j^{Yes} \cap X'|$ and that for roll $i \in E$ we have $|A_i^{Yes} \cap X| > |A_i^{Yes} \cap X'|$. Since $X' \in F_{Tezos}(\boldsymbol{A})$, F_{Tezos} is **not** Pareto efficient per Definition 4.1.

Therefore, we have proven that F_{Tezos} is **not** Pareto efficient per Definition 4.1 under the listed assumptions by showing that when $|N| > 0$ and $F_{Tezos} = \{\emptyset\}$, there will always exist a committee $X \in 2^P$ such that for all eligible-to-vote rolls $i \in E$: $X \succcurlyeq_{A_i^{Yes}} X'$ and for at least one roll $i \in E$: $X \succ_{A_i^{Yes}} X'$.

5.5 Non-Optimal Division

In this section, we examine the settings where community division becomes the most optimal outcome for the Tezos stakeholders. We achieve this by applying the work done in Section 3.5 to Tezos and its governance system. Namely, we apply Nash equilibrium to Tezos governance and simplify the governance process to a two-player strategy game.

Recall that every voting process in Tezos terminates with either a proposal being promoted or no change occurring. Given that each roll is treated as a single vote, we will refer to the set of rolls that vote as V, where $|V| = n$. We will assume that in each voting process:

- the participation ratio is 1 throughout all stages, and
- a proposal p passes the first stage of voting.

We define the individual actions of V's members, aggregated over the second and third stages of voting, as follows:

1. Promote the proposal p.
2. Reject the proposal p, resulting in no change to the ledger's protocol.

We refer to the broader community of stakeholders in Tezos as C, where $|C| = k$. We define the individual actions of C's members *in response to* the outcome of the voting process as:

1. Stay as a stakeholder in Tezos.
2. Leave Tezos.

We define β as the proportion of rolls, V, that voted to promote the proposal p and define γ as the proportion of all stakeholders, C, that leave Tezos in response to the outcome of the voting process.

Recall that in Section 3.5 we observed that there are four possible scenarios when proposal p is subject to a vote:

1. p is promoted:
   (a) The majority of C stays, or
   (b) The majority of C leaves.
2. p is rejected:
   (a) The majority of C stays, or
   (b) The majority of C leaves.

As we are interested in scenarios of community division, we decompose 1(b) and 2(b) into the following cases:

- **Case 1**: from 1(b). The majority of C wants p to be rejected and would leave if it is promoted, a *super-majority* of V votes to promote p and p is promoted. Thus, $\beta \in (0.8, 1]$ and $\gamma \in (0.5, 1]$.
- **Case 2**: from 2(b). The majority of C wants p to be promoted and would leave if it is rejected, a *majority* of V votes to promote p and p is rejected. Thus, $\beta \in (0.5, 0.8]$ and $\gamma \in (0.5, 1]$.
- **Case 3**: from 2(b). The majority of C wants p to be promoted and would leave if it is rejected, only a *minority* of V votes to promote p and p is rejected. Thus, $\beta \in [0, 0.5]$ and $\gamma \in (0.5, 1]$.

Like in [17], we use the software tools for game theory, version 15.1.1, provided by the Gambit project [47] to simulate a two-player strategy game for V and C. We use Gambit to compute the Nash equilibria for each case for the different values of β and γ.

Case 1

Let $\beta = 0.90$ and let $\gamma = 0.70$ (since $\beta \in (0.8, 1]$ and $\gamma \in (0.5, 1]$). The payoff matrix in Table 5.1 represents Case 1.

		The Broader Community (C)	
		Leave	**Stay**
The Voting Rolls (V)	**Promote**	0.90, 0.70	0.90, 0.30
	Reject	0.10, 0.70	0.10, 0.30

Table 5.1: The payoff matrix for the Case 1.

Observe that there is only one Nash equilibrium, which occurs in the grey cell. The Nash equilibrium lies with the event of 90% of V, that voted to promote proposal p, and 70% of C leaving Tezos. This result can be further generalised for any $\beta \in (0.8, 1]$ and any $\gamma \in (0.5, 1]$. In other words, whenever the majority of C believe their payoff is maximised in leaving Tezos if p is promoted and the super-majority of V vote to promote p, the optimal outcome is for the majority of V and the majority of C to leave Tezos. This does divides the Tezos community.

Case 2

Let $\beta = 0.79$ and $\gamma = 0.70$ (since $\beta \in (0.5, 0.8]$ and $\gamma \in (0.5, 1]$). The payoff matrix in Table 5.2 represents Case 2.

		The Broader Community (C)	
		Leave	**Stay**
The Voting Rolls (V)	**Promote**	0.79, 0.70	0.79, 0.30
	Reject	0.21, 0.70	0.21, 0.30

Table 5.2: The payoff matrix for the Case 2.

Observe that there is only one Nash equilibrium, which occurs in the grey cell. The Nash equilibrium lies with the event of 79% of V, that voted to promote proposal p, and 70% of C leaving Tezos. This result can be further generalised for any $\beta \in (0.5, 0.8]$ and any $\gamma \in (0.5, 1]$. In other words, whenever the majority of C believe their payoff is maximised in leaving Tezos if p is rejected and the majority of V vote to promote p but there is no super-majority, the optimal outcome is for that majority of V and the majority of C to leave Tezos. This divides the Tezos community.

Case 3

Let $\beta = 0.40$ and $\gamma = 0.70$ (since $\beta \in [0, 0.5]$ and $\gamma \in (0.5, 1]$). The payoff matrix in Table 5.3 represents Case 3.

Observe that there is only one Nash equilibrium, which occurs in the grey cell. The Nash equilibrium lies with the event of 60% of V, that voted to reject proposal p, and 70% of C leaving Tezos. This result can be further generalised for any $\beta \in [0, 0.5]$ and any $\gamma \in (0.5, 1]$. In other words, whenever the majority of C believe their payoff is maximised in leaving Tezos if p is rejected and only a minority of V vote to promote p, the optimal outcome is for the majority of V and the majority of C to leave Tezos. This divides the Tezos community.

Observations

From the analysis above, we have observed that whenever there is a majority of C that would leave Tezos in response to the outcome of the voting process, the optimal outcome would divide the Tezos community.

		The Broader Community (*C*)	
		Leave	**Stay**
The Voting Rolls (*V*)	**Promote**	0.40, 0.70	0.40, 0.30
	Reject	0.60, 0.70	0.60, 0.30

Table 5.3: The payoff matrix for the Case 3.

Since V represents C via delegation, scenarios 1(b) and 2(b) are only possible whenever:

- a minority of C possesses the majority of rolls and is opposed to the majority of C (Case 1 and 3), or
- a minority of C possesses at least 20% of the rolls to prevent a super-majority for p (Case 2).

For either scenario to take place, a large degree of stake centralisation is required. CoinMetrics [48] have shown that, in December 2019, 0.26% of the unique Tezos addresses held 82% of the total supply of tokens and 1.62% of the unique Tezos addresses held 96% of the total supply of tokens. Although each unique address does not correspond to an independent operator, the stake distribution is skewed enough to conclude that a minority of the stakeholders hold the majority of the available stake. And, since 80% of the stake is needed for a super-majority, this poses a substantial division risk to the Tezos community in the event of the majority of C being firmly opposed to the outcome of the voting process as per our analysis above.

We conclude that such a degree of centralisation is a condition that make it feasible for community division to be a Nash equilibrium, as we have demonstrated in Case 1, 2 and 3. As a result, Tezos does **not** meet the Non-Optimal Division property per Definition 5.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this work, we derived five desired blockchain governance properties from real-world examples, governance models, social choice theory and game theory. We have shown that Tezos does not offer private ballots, at the expense of offering verifiability, and that it does not directly incentivise participation. We have also proven that the Tezos governance system is not Pareto efficient per our derived definition due to its capability of terminating without a winning proposal. Finally, we have shown that, as of December 2019, the supply distribution in Tezos makes it feasible for community division to be a Nash equilibrium.

6.2 Future Work

In Chapter 3, we mentioned that we only chose to utilise *some* of the insights from Chapter 2 in deriving the desired properties. In particular, we only chose to utilise insights from real-world examples and the Treasury model [29]. We note that we could have utilised other insights from the other two models (Futarchy [26] and Quadratic Voting [27]); however, we chose to adopt those that we deemed most fundamental and practical. A future iteration of this work can further examine some of the insights from the other models as potential desired properties of governance. For example, it can be argued that capturing the intensity of voters' preferences (the main insight from Quadratic Voting) is fundamental to fair governance; however, its practical potential should be further examined.

Another potential quest for future work could be making the derived properties as general as possible, in order to be as applicable to as many distributed ledgers. For example, the Non-Optimal Division property (Section 3.5) could be altered to account for both self-amending and non-self-amending blockchains. Additionally, this property could be further expanded to predict the existence of the conditions that allow for optimal divisions to happen, instead of only looking at whether these conditions exist or not.

Finally, a more practical future iteration of this work could use the derived properties to build a distributed ledger or a decentralised organisation that satisfy each of the properties. This could then be tested to examine how beneficial the properties are in avoiding governance problems.

Bibliography

[1] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. White paper. Mar. 2009. URL: https://bitcoin.org/bitcoin.pdf.

[2] LM Goodman. *Tezos—a self-amending crypto-ledger*. White paper. 2014. URL: https://www.tezos.com/%20static/papers/white_paper.pdf.

[3] *CoinMarketCap*. URL: https://coinmarketcap.com/ (visited on 10/03/2020).

[4] Florian Tschorsch and Bjorn Scheuermann. "Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies". In: *IEEE Communications Surveys Tutorials* 18 (Mar. 2016), pp. 1–1. DOI: 10.1109/COMST.2016.2535718.

[5] Ralph C. Merkle. "A Digital Signature Based on a Conventional Encryption Function". In: *CRYPTO '87: A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology*. London, UK: Springer-Verlag, 1988, pp. 369–378. ISBN: 3-540-18796-0.

[6] Steven Goldfeder; Arvind Narayanan; Joseph Bonneau; Andrew Miller; Edward Felten. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.

[7] Gareth Peters and Efstathios Panayi. "Understanding Modern Banking Ledgers Through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money". In: Sept. 2016, pp. 239–278. ISBN: 978-3-319-42446-0. DOI: 10.1007/978-3-319-42448-4_13.

[8] J. Garay and A. Kiayias. "SoK: A Consensus Taxonomy in the Blockchain Era". In: *CT-RSA 2020: 284-318* (Feb. 2020).

[9] Ziyao Liu et al. "A Survey on Applications of Game Theory in Blockchain". In: *CoRR* abs/1902.10865 (2019). arXiv: 1902.10865. URL: http://arxiv.org/abs/1902.10865.

[10] S. King and Scott Nadal. *PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake*. White paper. 2012. URL: https://decred.org/research/king2012.pdf.

[11] The Nxt Community. *Nxt*. White paper. URL: https://www.jelurida.com/sites/default/files/NxtWhitepaper.pdf.

[12] Primavera de Filippi and Greg Mcmullen. "Governance of blockchain systems: Governance of and by Distributed Infrastructure". In: *Blockchain Research Institute and COALA* (2018). DOI: hal-02046787.

[13] Rowan van Pelt et al. "Defining Blockchain Governance: A Framework for Analysis and Comparison". In: *Information Systems Management* (Mar. 2020). DOI: 10.1080/10580530.2020.1720046.

[14] Aggelos Kiayias, Andrew Miller, and Dionysis Zindros. "Non-interactive Proofs of Proof-of-Work". In: July 2020, pp. 505–522. ISBN: 978-3-030-51279-8. DOI: 10.1007/978-3-030-51280-4_27.

[15] Arie Kapteyn. "Utility and economics". In: *De Economist* 133 (Jan. 1985), pp. 1–20. DOI: 10.1007/BF01675959.

[16] Cristina Bicchieri. "Rationality and Game Theory". In: Feb. 2004, pp. 182–205. ISBN: 9780195145397. DOI: 10.1093/0195145399.003.0010.

[17] Nida Khan et al. *Blockchain Governance: An Overview and Prediction of Optimal Strategies using Nash Equilibrium*. 2020. arXiv: 2003.09241 [cs.GT].

[18] Kiayias A. et al. "Updatable Blockchains". In: *Computer Security – ESORICS 2020* (Sept. 2020). URL: https://doi.org/10.1007/978-3-030-59013-0_29.

[19] Benito Arruuada and Luis Garicano. "Blockchain: The Birth of Decentralized Governance". In: *SSRN Electronic Journal* (2018). DOI: 10.2139/ssrn.3160070.

[20] Vitalik Buterin. *Ethereum White Paper*. White paper. URL: https://ethereum.org/en/whitepaper/.

[21] Divisions of Corporation Finance and Enforcemen. *Statement by the Divisions of Corporation Finance and Enforcement on the Report of Investigation on the DAO*. Investigation report. July 2017. URL: https://www.sec.gov/litigation/investreport/34-81207.pdf.

[22] Evan Duffield and Daniel Diaz. *Dash: A payments-focused cryptocurrency*. White paper. 2018. URL: https://github.com/dashpay/dash/wiki/Whitepaper.

[23] Haris Aziz and Nisarg Shah. *Participatory Budgeting: Models and Approaches*. 2020. arXiv: 2003.00606 [cs.GT].

[24] D. Kaidalov, L. Kovalchuk, and et al. "A proposal for an Ethereum Classic Treasury System." In: *IOHK* (Mar. 2017).

[25] *Dash Github Repository*. URL: https://github.com/dashpay/dash (visited on 01/15/2021).

[26] Robin Hanson. "Shall We Vote on Values, But Bet on Beliefs?" In: *Journal of Political Philosophy* 21.2 (2013), pp. 151–178. DOI: 10.1111/jopp.12008. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jopp.12008. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/jopp.12008.

[27] Steven P. Lalley and E. Glen Weyl. "Quadratic Voting: How Mechanism Design Can Radicalize Democracy". In: *AEA Papers and Proceedings* 108 (2018), pp. 33–37. DOI: 10.1257/pandp.20181002.

[28] Juan Huertas, Hope Liu, and Sarah Robinson. *Eximchain: Supply Chain Finance solutions on a securedpublic, permissioned blockchain hybrid*. White paper. URL: https://eximchain.com/static/media/Whitepaper-Eximchain.c26dcb47.pdf.

[29] Bingsheng Zhang, Roman Olynykov, and Hamed Balogun. "A Treasury System for Cryptocurrencies: Enabling Better Collaborative Intelligence". In: *NDSS* (2019).

[30] Charles Hoskinson. *Cardano*. White paper. URL: https://whitepaper.io/document/581/cardano-whitepaper.

[31] Ari Juels, Dario Catalano, and Markus Jakobsson. "Coercion-Resistant Electronic Elections". In: vol. 6000. Jan. 2010, pp. 37–63. ISBN: 978-3-642-12979-7. DOI: 10.1007/978-3-642-12980-3_2.

[32] Chaieb Marwa et al. "Verify-Your-Vote: A Verifiable Blockchain-Based Online Voting Protocol: 15th European, Mediterranean, and Middle Eastern Conference, EMCIS 2018, Limassol, Cyprus, October 4-5, 2018, Proceedings". In: Jan. 2019, pp. 16–30. ISBN: 978-3-030-11394-0. DOI: 10.1007/978-3-030-11395-7_2.

[33] Édouard Cuvelier, Olivier Pereira, and Thomas Peters. "Election Verifiability or Ballot Privacy: Do We Need to Choose?" In: *Computer Security – ESORICS 2013*. Ed. by Jason Crampton, Sushil Jajodia, and Keith Mayes. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 481–498. ISBN: 978-3-642-40203-6.

[34] Wolter Pieters. "Verifiability of electronic voting: between confidence and trust". Undefined. In: *Data Protection in a Profiled World*. Ed. by Serge Gutwirth, Yves Poullet, and Paul De Hert. 10.1007/978-90-481-8865-9$_9$. Springer, 2010, pp. 157–175. ISBN: 978-90-481-8864-2. DOI: 10.1007/978-90-481-8865-9_9.

[35] David Chaum. "Secret-Ballot Receipts: True Voter-Verifiable Elections". In: *Security Privacy, IEEE* 2 (Feb. 2004), pp. 38–47. DOI: 10.1109/MSECP.2004.1264852.

[36] Tassos Dimitriou. "Efficient, Coercion-free and Universally Verifiable Blockchain-based Voting". In: *Computer Networks* 174 (2020), p. 107234. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2020.107234. URL: http://www.sciencedirect.com/science/article/pii/S1389128619317414.

[37] Renee Irvin and John Stansbury. "Citizen Participation in Decision Making: Is It Worth the Effort?" In: *Public Administration Review* 64 (Feb. 2004), pp. 55–65. DOI: 10.1111/j.1540-6210.2004.00346.x.

[38] Bryan Caplan. "Rational Ignorance". In: vol. 74. Jan. 2003, pp. 793–795. DOI: 10.1007/978-0-306-47828-4_170.

[39] Boas Kluiving et al. "Analysing Irresolute Multiwinner Voting Rules with Approval Ballots via SAT Solving". In: *ECAI* (2020).

[40] Ronald Rivest and Emily Shen. "An Optimal Single-Winner Preferential Voting System Based on Game Theory". In: *Proc. of the 3rd Intl. Workshop on Computational Social Choice (COMSOC)* (Aug. 2010).

[41] Piotr Faliszewski, Arkadii Slinko, and Nimrod Talmon. "The Complexity of Multiwinner Voting Rules with Variable Number of Winners". In: (2017). arXiv: 1711.06641 [cs.GT].

[42] Jerry S. Kelly. "Strategy-proofness and social choice functions withoutsingle-valuedness". In: *Econometrica, 45(2):439–446* (1977).

[43] *Tezos Gitlab Repository*. URL: https://gitlab.com/tezos/tezos/-/tree/master (visited on 02/20/2021).

[44] Michael Neuder et al. *Defending Against Malicious Reorgs in Tezos Proof-of-Stake*. 2020. arXiv: 2009.05413 [cs.CR].

[45] Michael Neuder et al. *Selfish Behavior in the Tezos Proof-of-Stake Protocol*. 2020. arXiv: 1912.02954 [cs.CR].

[46] Jonah Brown-Cohen et al. “Formal barriers to longest-chain proof-of-stake protocols”. In: *Proceedings of the 2019 ACM Conference on Economics and Computation*. ACM, 2019, pp. 459–473.

[47] R. D. McKelvey, A. M. McLennan, and T. L. Turocy. *Gambit: Software Tools for Game Theory*. URL: `http://www.gambit-project.org` (visited on 03/31/2021).

[48] The Coin Metrics Team. *Analyzing the Supply Distributions of Projects with On-Chain Governance*. Report. URL: `https://coinmetrics.substack.com/p/coin-metrics-state-of-the-network-768`.