A Technical Perspective on Decentralization

By Craig Calcaterra and Wulf Kaal

Abstract

This chapter presents a technical perspective on decentralized structures. The purpose of the chapter is to provide guideposts and an overview of the technical possibilities and limitations for the creation of decentralized autonomous organizations (DAOs). A core precept for the creation of DAOs is the recognition that any set process or set of rules that can ever be designed will ultimately fail to secure a network for all time. This insight is particularly relevant for a decentralized network that should be open for anyone on the globe and provide anonymity. How can a DAO be governed with a set of rules that always need to be amended in the future as the environment changes, to keep the network running healthily? It is necessary to design the systems with an evolutionary mindset and a dynamic governance process. The chapter shows how a dynamic environment for the governance of DAOs can be created and how DAOs may thrive with such a design.

The book can be accessed here:
https://www.amazon.com/Decentralization-Technologies-Organizational-Societal-Structure/dp/3110673924/
and here:
https://www.degruyter.com/view/title/569051

Chapter 4. Technical perspective

Before we explain how to build the major institutions that are lacking in the decentralized economy, we need the technical perspective to explain why these particular solutions are even necessary. Setting up the rewards and punishments in a DAO to ensure productive collaboration is not easy without a central authority to umpire disagreements and maintain harmony. Impossibility results illustrate the right perspective for developers to understand the logical limits of democratic collaboration. Game theory is necessary to properly justify incentive design.

Since this is not a technical book, we assume our readers have a variety of different technical backgrounds—including nearly none. However, authors don't truly understand a subject if they can't explain it simply, without technical jargon, to an intelligent and interested audience. Nevertheless, this is a technology subject, so we need to make a short excursion and explore some technical ideas, to put the leading-edge solutions that programmers are developing in perspective.

Our goal is to build a DAO, a decentralized autonomous organization, which has open membership to anonymous individuals from around the globe. In this chapter we argue that building effective and efficient DAOs requires a secure and meaningful reputation system, maximum bureaucratic transparency through a dynamic governance structure, and coherent transcendental values for long-term stability.

Building such large decentralized networks is a new challenge in history. Previously, groups were unified by local identifiers. At the very beginning, people would share close family. Then religions unified many different families. Then notions of patriotism, culture, and philosophy bonded people who shared cultural similarities such as languages. Geographical closeness could unite people into kingdoms by their similar experiences and lifestyles. Ancient Mesopotamia and China were able to unify many diverse cultures and geographies thanks to the advance in information technology of writing and increasingly complex ideas (bureaucracy and laws). When the laws were strict—Rule by Law—then the hierarchy formed rigidly and was very efficient temporarily, before its rigidity led to its instability and eventual collapse. This is protocol centralization. When the laws were loose—Rule by Virtue (throughout most of Ancient Egyptian and Chinese history)—then the organization was more stable. This is protocol decentralization.

Later, modern Western democracies with diverse populations required the unifying force of rigid protocol centralization. These letter-of-the-law legal systems were required to achieve political decentralization, since equality demands impartial and universal application of the law. To maintain stability under this rigid protocol centralization, the founders instituted a dynamic political system. Its concrete rules for shifting power periodically and building checks and balances into the rigid protocol relieves the tension of the impersonal bureaucracy and rigid power hierarchy.

Our contemporary challenge is to bring together people of every background who wish to contribute toward an economic goal. Ideally, members could participate anonymously, to maximize the size of the network and encourage contributions. With these obstacles, how can we possibly create a system that stays coherent?

The new tools of information technology allow rules with perfect logical rigidity—rules which are rigorously and immediately enforced by smart contracts. Their application may be completely transparent for everyone in the global network to witness. This allows unprecedented protocol centralization, which will help keep such diverse

groups organized and coherent as they strive to cooperate toward their economic goals, while remaining politically decentralized.

Despite the challenges exposed by rigid smart-contract-executed protocols that govern DAOs, we also have the tools to address these challenges. Contemporary information technology allows us to decentralize power by polling every member nearly instantaneously. Our processing ability allows any regulating instructions to be securely computed nearly instantaneously, without a central executive power. Our decentralized information storage and processing tools allow us to keep track of contributions and to reward and punish behaviors fairly without any ultimate, centralized arbiter to resolve disputes.

However, protocol centralization leads to instability if it is not implemented wisely. The point of this technical chapter is to illustrate two things:

No static set of rules can ever perfectly reflect the will of the group without leaving loopholes for adversaries to profit at the expense of the majority. There will always be an arms race between policing and those who will push strict rules to the limit. Therefore, to achieve stability a DAO must institute a dynamic governance system with a clear and accessible processes for amending the rules and appealing the automated conclusion of smart contracts.

Since decentralized organizations don't have leaders or a hierarchy of control, any governance process must be instituted from the very beginning.

Proper incentives are crucial for harmonious collaboration.

Short-term business deals require a secure and meaningful reputation system.

Long-term stability is determined by transcendental values.

To illustrate the logical necessity of these stipulations, in this chapter we briefly summarize some mathematical results from economics and social science: Condorcet's paradox, Arrow's impossibility theorem, and the Folk Theorems of Game Theory.

Impossibility results

There is a cultural difference between mathematicians and engineers which meets at the boundary in computer science. Engineers are trained to believe that for any problem in any situation, we can find a solution. Given any gorge, we can design a bridge that can safely span the distance.

Mathematicians, on the other hand, are trained to categorize and completely understand the set of all possibilities related to a problem, and outline what is possible and impossible given a strictly delimited set of rules. This leads them to anticipate that there are questions which have no solutions. Math is filled with impossibility results. Some very basic examples:

$\pi = 3.14159 \ldots$ is irrational, meaning it is impossible to write the number as a fraction with two integers.

It is impossible to trisect some angles with straightedge and compass.

It is impossible to find an algebraic formula for solving general 5th order polynomial equations.

There is no elementary antiderivative for the Gaussian distribution.

There are literally hundreds of impossibility results in the computer science field of distributed computing.1

This does not mean these problems cannot be solved. These results just show you cannot solve the problems within the limitations of a fixed set of tools. The engineering attitude that anything can be solved may be valid, if you look at the problem from a new angle and invent a new strategy.

The intuition we are trying to convey in this chapter is that there are logical limits to what can be built. If you don't respect them, you will bang your head against an unsolvable problem. The impossibility results show we can never design a static set of rules which will eternally sustain a DAO. But that just means we need to set up DAOs to run from the beginning with a dynamic set of rules.

Condorcet's paradox

Marquis de Condorcet was a philosopher and mathematician who embodied the ideals of Enlightenment rationalism during the French Revolution in the 18th century. Condorcet promoted free and equal public instruction and equal rights for women and people of all races. As the main author of the Girondin constitutional project, he built these ideas explicitly into the primary rules of his political party's proposed French Constitution.2

However, Condorcet's constitution was never put to a vote. The Montagnards asserted their own rules, which became the French Constitution of 1793, after they gained control of the convention. Condorcet criticized their proposal and was named

[1] Hagit Attiya and Faith Ellen, *Impossibility Results for Distributed Computing*, Morgan & Claypool (2014)

[2] *Plan de Constitution présenté à la Convention nationale les 15 et 16 février 1793, l'an II de la République (Constitution Girondine)* is an elegant work which contains designs which are still relevant to building DAO constitutions today. For example, the plan was to cycle the presidency of the executive council between 7 ministers every fifteen days. Elections would occur in two stages. Candidates consist of the members with the most votes in the first stage, so that any candidate in the population has the opportunity to be chosen. Three times as many candidates as the number of open seats then run during the second stage of voting.

a traitor. The Montagnards were the most radical group in the Assembly. They were responsible for the Reign of Terror under their leader, Robespierre. In the course of one year 16,594 official death sentences were carried out and an additional 10,000 died in prison without trial, including Condorcet himself.3

Before he was imprisoned, Condorcet continued to promote Enlightenment ideals while in hiding, by writing what has been described as the final word on the Enlightenment.4 Sketch for a Historical Picture of the Progress of the Human Spirit (1795) argues that progress in the history of civilization is measured by improvements in justice, which are achieved in step with our advances in scientific understanding of the world.

While he believed the goal of humanity was to strive toward ever more just and productive societies which increase our individual potentials, Condorcet was not a naive utopian scientific rationalist. Condorcet was a subtle thinker. His impossibility result, discussed next, illustrates how a perfect mathematical/mechanical governance process is not feasible. Condorcet subsequently focused on promoting the incongruous qualities of diverse individuality as the best means for improving the morality and justice in our social structures. Diverse humanity is Condorcet's foundation for objectively superior values. Universal and eternal principles are impossible for humanity to apprehend. Individual liberty is therefore crucial for a society to improve their path through history.

Condorcet's paradox5 demonstrates it is impossible to construct any method that will faithfully discover the will of a group, with any type of democratic voting system. Whenever there are 3 or more candidates, there can be circularities of preference, similar to the paper-rock-scissors cycle, with no clear winner. This simple observation proves it is not possible to design a perfect democratic governance system.

Here three different majorities within the group might prefer A over B, B over C, and C over A. For example, see the following preference table for three voters:

Voter 1	Voter 2	Voter 3
A	B	C
B	C	A
C	A	B

Table 1: Voter preferences

[3] Condorcet was in prison at the same time as the American revolutionary Thomas Paine, who also coauthored the Girondin constitution.

[4] David Williams, *Condorcet and Modernity*, Cambridge University Press (2004)

[5] William Gehrlein, "Condorcet's paradox and the likelihood of its occurrence: different perspectives on balanced preferences", Theory and Decision. 52 (2): 171–199, (2002)

The result is:

2 voters prefer A to B.

2 voters prefer B to C.

2 voters prefer C to A.

Even though each individual has a clear personal ranking of the candidates, the group itself may have no clear ranking of what is best.

We may think that the situation fabricated in Table 1 is rare, but in fact the probability of a cycle occurring when there are 3 candidates is higher than 8% when there are more than 10 voters. As the candidate options grow, the chances of a cycle occurring somewhere amongst the options quickly rises to 100%.

One of the dangerous consequences of this situation is in primary voting. Consider the situation where A and B are paired in a primary before the winner runs against C in the general election. Under the preferences chosen in Table 1 we see A beats B in the primary, then A loses to C in the general election. Candidate C wins overall. But rearranging who runs first in the primary can force any of the candidates to win. Therefore, in this situation, those who have power over arranging the order of the contest, have the ability to determine the result. This is especially an issue with sports tournaments, as the organizers have some power in determining the outcome based on their choice of initial matchups.

Application to network forking

These Condorcet cycles are the simplest demonstration that no rules can be created to consistently discover the will of the group. But the astute reader may object, that in this situation, the group doesn't really have a preference, so we can't expect any process to conclude otherwise. First, that is a major problem for democracy. Sometimes there is no consensus to be had. This leads to the second conclusion that sometimes network forking is inevitable.

A network fork happens when two or more subgroups of the network split into separate networks. The profusion of religious sects provides voluminous historical examples. The most famous example in blockchain was when the Ethereum network experienced an irreconcilable philosophical difference early in its history with the original 2016 DAO.

In 2016, a decentralized venture capital fund, the DAO, was crowdfunded with a US$120 million token sale. One month later a third of its funds were siphoned off with an unexpected programming exploit. The split in the Ethereum community came down to two opinions. The majority of members chose to refund the money to the investors in the young network by changing the software that the members employ to communicate with each other. A sizeable minority refused to switch to the new

software, continuing as before, changing the name of their fork of the network to Ethereum Classic.

In general, forking is bad, and forking is good. When a network splits in half, from an abstract perspective, we consider that each side is worth one quarter of its previous value or power. That's bad. The idea is that a network is valuable because of its connections. A general rule of thumb is to calculate the power of a network by squaring the number of members. A network with twice as many members has 4 times as much power.6 This is the primary quality referred to with the term "network effects".

However, forking is probably good in the long term. Persistent, profound differences in goals and talents can arise within a group. This signals the need for specialization of domains of power. To be able to achieve that specialization without splitting irrevocably in a hard fork is sometimes more efficient as it keeps the strength added by more members and connections, but not always if the member-multiplier is outweighed by the strife involved with forcibly maintaining a bond between members with irreconcilable approaches to solving a problem.

The general problem of coming to democratic consensus is much worse than the mere existence of cyclic voting preferences. Even if we ignore these cyclic situations, there are still more elementary logical obstacles to deciding issues fairly. Another basic voting result to be concerned about is Duverger's law, which applies to the current American system for deciding the president. The official ranking system is called plurality voting: whoever gains the most votes (the plurality) wins. This has long been understood to lead to our current two-party system. Third parties are unstable, since people naturally want their vote to matter if they bother to vote. Therefore, under plurality voting, people are naturally incentivized to game the system by voting for their preference of the perceived top two candidates, instead of "wasting their vote" on someone who has less chance—even if a third person is their authentically preferred candidate. Small parties never win, so there tends to be only two viable parties in the long run. These two parties must distinguish themselves from each other. So plurality voting predictably leads to a system which swings between two polarized parties who are less acceptable to the majority, but strongly preferred by interested minorities—Duverger's Law.

[6] Is this theoretical claim backed up by data? In this example we come to no conclusion. In general, it is no surprise that Ethereum lost some value in its market cap in the months after the DAO event. But there are many variables in play. The price of this very young network was (and continues to be) extremely noisy. So, it's not even definitively clear the event affected its market cap in any particular direction, in the long term. The failure of the network could account for more of the drop than the split. Or the quick and decisive response could account for the fact that it didn't collapse. Or that same response is interpreted by some (including those who split) as reason for the drop.

Such obstacles to democracy are part of the reason it can be more efficient to rely on dictatorships and centralization, for example in private companies. We will advance through more complex results in the next sections.

Arrow's Impossibility Theorem

In 1951 Kenneth Arrow proved that there is no voting system which decides the winner of an election according to the will of the voters in all situations. Arrow makes some simple, reasonable assumptions, then proves they cannot all be met under one system.

Stated briefly, Arrow's Impossibility Theorem proves it is not possible to design any system, within specific strict static assumptions, which a decentralized organization (a group without a dictator) can follow to come to consensus on a question with 3 or more options in a fair way. "Fair" is defined according to the technical, but reasonable, requirements of Unrestricted Domain, Pareto Efficiency, and Independence of Irrelevant Alternatives.

Unrestricted Domain means each individual voter has the freedom to choose any ordered ranking of the candidates from first to last.

Pareto Efficiency means if every member of the group prefers candidate A to B, then the system cannot choose candidate B as the winner.

Independence of Irrelevant Alternatives means that if the system would choose A as winner, the system should not change the winner just because a single voter who prefers A > B > C might change their vote to preferring A > C > B. In this case, B and C are the "irrelevant alternatives".

Arrow's Impossibility Theorem proves you can't have all three assumptions in one system. This shows that if you want a democratic system with very basic standards for fairness (Unrestricted Domain and Pareto Efficiency), there will always still be an opportunity for voters to disrupt the election with strategic voting (manipulating their irrelevant alternatives). A sufficiently patient and clever minority power can always corrupt the process and profit at the expense of the majority, assuming your process has static rules and finite, discrete execution. There is no perfect voting system. All processes can be manipulated and corrupted while following the rules.

No voting method can be constructed to decide the winner of an election that properly reveals the preferences of two or more voters on three or more candidates, which satisfies these basic and obviously desirable assumptions. For instance, the most complicated axiom, independence of irrelevant alternatives, requires the winner should not change if a voter changes their opinion about the relative ranking of two losers.

Similarly, our much more complex goal of finding a single, automated system which correctly rewards all contributors to a collaboration can never hope to be perfect. Instead such results encourage us to widen our perspective. We analyze families of different systems to determine which motivations are incentivized by which reward systems. Then we ensure the system is motivated to dynamically respond to the inevitable gaming and attacks that will occur, from within and without, whenever money can be won, with an evolutionary protocol for continual improvement.

The natural question is whether we can use some other polling method to prevent the strategic subversion of the intentions of the majority with good will. Can we police strategic voting, punishing false reports? We have new tools for voting now—new technologies for communication and recording. We can weight our preferences continuously, splitting our vote into percentages between candidates. We can know the state of the election and change our vote continuously up to the last minute. We can coordinate with other voters while monitoring the poll. Does that help or hurt discovery of the will of the group?

Many alternatives have been explored in the literature. In all the methods discussed the answer has always been that changes both help and hurt. New approaches solve old problems, but they create new opportunities for manipulation. In computing this is called “increasing the attack surface”. To give us a better intuition for the theoretical limits of our modeling abilities, infinitely many more alternative solutions—and problems—are illustrated in the next section.

Folk Theorems of Game Theory

The Folk Theorems of Game Theory are important for designing decentralized organizations, because they rigorously illustrate two ancient dictates of common wisdom.

Delayed gratification is crucial for success.
The incentive structure is radically changed between short-term and long-term perspectives. A secure system for tracking meaningful reputation is crucial for creating long-term stability when designing a DAO. “A good reputation is more valuable than money.”7

The spirit of the law is more important than the letter of the law.
It is impossible to set up formal rules for a group to keep them behaving in a cooperative manner in the long run, even if the rules are policed perfectly. Evolution must be

[7] Publilius Syrus (85—43 BC), Sententiae, Maxim 108. Cf., 大器晚成 Large vessels take longer to complete. Common Chinese idiom/chengyu.

built into the design of any temporary rules. To keep these dynamic rules coherent in the long run, a decentralized organization must commit to transcendental values as primary. “Transcendental” means these values cannot be specified precisely, logically, and completely with formal rules. 8

The Folk Theorems give a rigorous justification of these two aphorisms, so that technological constructions such as digital DAO constitutions can be engineered carefully for long-term stability.

The goal of this section is to explain the Folk Theorems of Game Theory and how they apply to network situations, as clearly as possible to a general audience who is not interested in examining the minutia of every mathematical detail.9

Prisoner’s Dilemma

To illustrate the Folk Theorems, let us go through the most famous basic example of a strategic game called the Prisoner’s Dilemma (PD). Two people get together to exchange closed bags, with the understanding that one of them contains money, and the other contains diamonds. Either player can choose to cooperate by putting their assets into their bag, or they can defect by handing over an empty bag.10

Table 2: https://en.wikipedia.org/wiki/Prisoner%27s_dilemma

SELLER / BUYER	COOPERATE	DEFECT
COOPERATE	1 1	2 -1
DEFECT	-1 2	0 0

Table 3: Reward matrix

[8] “When mores are sufficient, laws are unnecessary; when mores are insufficient, laws are unenforceable.” — Émile Durkheim
“Look, that's why there's rules, understand? So that you think before you break 'em.” —Terry Pratchett, *Thief of Time*, 2001.

[9] For these details, see, e.g., George J. Mailath and Larry Samuelson, Repeated Games and Reputations: Long-Run Relationships, Oxford University Press, 2006. This text is recommended because it highlights the case of repeated games with reputation.

[10] Paraphrased from Douglas Hofstadter, "Ch.29 *The Prisoner's Dilemma Computer Tournaments and the Evolution of Cooperation*", Metamagical Themas, Bantam Dell Pub Group, 1985.

Playing this game once gives only one stable and successful strategy: both players' best choice is to defect. This leads them to suffering the loss of opportunity of a good business deal, but not as much as if they lost their property. The strategy of defecting is best possible under the assumption that your adversary is brilliant (they can anticipate your strategy11) and ruthless (they will hurt you if it benefits them). This assumption is called rational self-interest in economics and game theory. The pair both defecting is what is called a Nash equilibrium, because any deviation from this strategy would lead to a worse outcome for one of the players, so eventually everyone will agree and settle at the equilibrium as the best strategy for the game.

It is worth developing a rigorous intuition for this situation. When we first encounter this game, our natural human instinct is to think of the defection strategy as cheating. It's bad. And we think of the cooperation strategy as self-evidently good, on a moral level. Why then do the game theorists insist the best strategy is to defect? Are game theorists morally bankrupt? Not exactly. They are looking at the game from a coldly calculating perspective of ignoring morality and merely accounting for what is the most efficient strategy. This perspective is useful for explaining why certain behaviors are observed in business and society, as the most efficient strategies tend to win out in the long run, despite moral misgivings.

If you are like most people, that will not assuage you. But the good news is, we can prove that cooperation is the optimal strategy if you allow people to play the game repeatedly—the moral perspective is the right and rational one if you have to live with the consequences of your behavior tomorrow. However, the Folk Theorems show that nothing is ever simple, and introduce a nagging wrinkle into any complex game.

Repeated Prisoner's Dilemma

Next, if we change the setup to assume the business deal is repeated day after day, the best strategy Nash equilibrium changes dramatically. The opportunity to repeatedly make a good business deal means both players would greatly improve their outcomes if they can find a strategy of long-term cooperation.

[11] It is often assumed that your adversary has mystical insight into your psychology. Axel Boldt (private communication) has another interpretation: Imagine both players have spies that can report back the strategies of their adversaries. Before you finalize a strategy and play the game, the spies report and so the players can change their strategies. But then the spies report on the new strategies. This can happen infinitely often before the game is finally played.

The first strategy is called grim trigger: you will cooperate as long as your adversary cooperates. If the adversary ever defects, then you promise to defect ever after, punishing your adversary forever.

Under the grim trigger strategy, your adversary's best option is to always cooperate. Any deviation from cooperation leads to a worse outcome, so eternal cooperation is a Nash equilibrium under the circumstances.

Grim trigger illustrates that the Nash equilibrium for a single stage of Prisoner's Dilemma (both players defecting) changes to a radically less efficient outcome if you are in a repeated game. Being stuck in the grim trigger trap is significantly worse, so the better strategy is to cooperate. Switching your perspective from a single point in time, to viewing all eternity in the future, we achieve a completely different perspective on our incentives. When we inhabit infinity, we realize our reputation for cooperation overrules our immediate impulses of fear and greed.

This new leap in efficiency as the players cooperate can only be achieved if there is the possibility of creating a policeable reputational system. The players must have a history, and that history must be available to the other players. Therefore, transparency and communication is essential for making policing effective and efficient.

Repeated Prisoner's Dilemma with precommitments

Another important insight that arises from repeated games is that there are infinitely many other strategies in this scenario which are also Nash equilibria. It is wrong to assume that there is some ideal strategy in any realistic scenario. This is a basic example of what is called a Folk Theorem in repeated games.

To illustrate this point, consider a new situation where we assume a competitor is allowed to precommit to a strategy. An example of a precommitment would be if you sign a smart contract which will automatically execute your orders in the future in a way that you cannot stop. Specifically, in the Noble vs. Peasant repeated Prisoner's Dilemma game set up, let's say the noble signals their precommitment to the strategy of cooperating 9 times out of 10 but deviating on the 10th time. The noble also precommits to the grim trigger strategy of defecting for all eternity if the peasant ever defects even once. Given this signal from the noble, the peasant now has the choice of defecting forever and gaining nothing, or cooperating forever and being betrayed 1 in 10 times, but profiting 9 in 10 times. The second option is more profitable for the peasant. So even though it is not fair (the noble gains +12 every 10 stages and the peasant gains +8), it is the rational choice for the peasant ($+8 > 0$.)

The Folk Theorem for this situation shows there are infinitely many different possible payout profiles the noble can force with different precommitment strategies. The noble can defect 4 times out of 10 to force a payout of $(noble, peasant) = (+14, +2)$.

The peasant still is better off with a payoff of +2 every 10 rounds instead of +0 for all eternity. Or the noble can defect 49 times out of 100, or 499 times out of 1000. The noble defecting anything less than 50% of the games still makes it profitable for the peasant to participate instead of suffering the consequence of the grim trigger for all eternity.12

Similar game setups with reputation policing explain the situation where a business will rationally tolerate a small level of theft when policing is more expensive, such as when a grocer will ignore a street urchin who nabs an apple once per day. This helps us understand the rationality of some exploitative long-term relationships such as parasites and bullying, and the need for careful protocol design to prevent that type of corruption/inefficiency.

This leads to the intuition that however rules are formalized in a realistic situation, there are strategies which follow the rules, but still subvert the intentions of the framers. There are legal strategies that profit the individual adversary at the expense of the group.

Infinite variations of Prisoner's Dilemma

The many folk theorems of game theory show there are many new strategies possible with each new assumption about the game. We showed this above with the change to repeated games and again with the change to allow precommitments. Each new change in assumptions leads to new strategies. This suggests there is no way to create a static protocol which can guarantee consensus behavior from all members of a group playing a complicated and realistic repeated game.

For example, there are new equilibria strategies when there are different levels of patience in the players (technically called discounting). Then old equilibria fail and new equilibria arise when you change the game to assume players have imperfect information. With each change of assumption in the game, old successful strategies fail to be Nash equilibria, and new successful strategies arise.

Or if you allow periodic opportunities for renegotiation, then you might be able to get out of a grim trigger trap. This sounds like it would improve outcomes, but in fact it gives new opportunities for defecting/cheating. If you know there is a chance

[12] Technically, for simplicity, we are assuming a discount factor of $\delta = 1$ so we can speak freely without extra qualifications. We can make this more complicated by modeling how patient the peasant is, $0 \leq \delta \leq 1$ but the wider conclusion stands.

to renegotiate, it might be to your advantage to defect and gain the reward, and gamble that you can talk your way out of the punishment of grim trigger later. Again, the Folk Theorem in this case reveals infinite successful strategies.

Further prisoner's dilemma alternatives that give new successful strategies include

partial or periodic opportunities for player communication (e.g., to punish shirkers, or collude, or reward players at retirement)

accounting for the cost of policing deviations

accounting for a player's history or reputation

stochastic variations, including

results are not perfectly reported

memory of the past is not perfect

strategies not implemented perfectly, "trembling hand"

new players allowed to enter or leave at various times, "long- or short-lived players"

tournaments

tournaments with history/reputation

changing assumptions about the population/market (e.g., how cooperative or ruthless)

evolutionary concepts added to incentives, e.g., accounting for a large population through time

continuous strategies (variable amounts of cooperation or betrayal)

asymmetric rewards (one party stands to lose or gain more than the other)

asymmetric timing (one party plays before the other)

Etc., etc., etc. All of these basic assumptions listed above apply to many different models of realistic situations. None of those models comes close to completely encapsulating the set of all behaviors that can occur in any natural setting.

We further argue this important point with the following meta-theorem13 due to the instability of Nash equilibrium strategies with respect to changing assumptions:

[13] We call this a meta-theorem because it can't be formalized. To make it mathematically rigorous, you would need to explicitly state the context or domain in which each successive subset is the domain of a subgame. E.g., if the action space was in $\mathbb{R}^n$ then we could imagine generalizing the game to $\mathbb{R}^{n+1}$. But the point of the meta-theorem is that whatever universe you specify as the domain, you can easily make a natural assumption that runs out of that category, e.g., from $\mathbb{R}^n$ to a probability space, and from there to a much more general metric space. And there are no limits to generalizing categories beyond that. (Basic paradoxes, such as Russell's paradox demonstrate the logical futility of posing the existence of "the set of all sets" as your universe.)

No matter which game you are playing, you can add a new assumption to make a more complicated generalization of the game, so that the previous winning strategy becomes a losing strategy, and a new winning strategy arises.

Life is not constrained within any given game's assumptions. You can always come up with a new strategy that takes advantage of the previous myopia under a static set of rules.14

Game theory gives us the intuition that in any slightly sophisticated repeated game (such as Prisoner's Dilemma), there is not one optimal strategy, but many possible successful strategies. Further, these successful strategies are unstable when new assumptions are made. Whenever we add a new twist to the formal description of a game, new successful strategies arise and old strategies become unsuccessful. Any realistic situation has innumerable assumptions. In fact you can always change the assumptions by behaving in a new way as you play any realistic game according to a new desire. If you design a network to improve its members' circumstance by cooperation, then if it succeeds, it will change their circumstance, and their desire. It is not possible to design a game in real life with a unique and perfect optimal strategy. In any realistic situation, people always discover new strategies or merely arrive at alternative successful behaviors which subvert the intentions of the designers.

Any set process or set of rules that we can ever design will ultimately fail to secure a network for all time. Especially a decentralized network. Especially if we want to allow anyone on the globe to join. Especially if we want to allow anonymity. Any set of rules that can ever be designed will always need to be amended in the future as the environment changes, to keep the network running healthily. We cannot rely eternally on a static set of rules, or else the system will inevitably become corrupt or irrelevant.

[14] This is the sort of perspective that Byzantine Fault Tolerance algorithm testers display. Byzantine behavior means breaking the stated rules. "(i) Byzantine behavior is unconstrained, hence, one can only implement a subset of such behaviors; and (ii) the subset of Byzantine behaviors to be tested are chosen by system developers, who are naturally tainted by having designed the system with certain limited Byzantine behaviors in mind." --Shehar Bano, Alberto Sonnino, Andrey Chursin, Dmitri Perelman, Dahlia Malkhi, "Twins: White-Glove Approach for BFT Testing" 22 April 2020 https://arxiv.org/pdf/2004.10617.pdf retrieved 5/6/20.
The point is that life cannot be entirely contained in any mathematical model, no matter how complicated. We have infinite-dimensional spaces that contain only a tiny fraction of a model. For example, if you try to model something as simple as a child's bouncy ball, you would need infinite dimensions to model its vibrational modes as an elastic sphere. But that is a hopelessly simplistic representation of an actual ball, as no actual ball is a perfect sphere. Every imperfection is ultimately important, but even if we started to include a few of the infinitely many imperfections, the model would quickly become incalculably difficult to render in a computer. So we simplify all scientific models drastically, so we can achieve computable results in a reasonable time.

The point of the Folk Theorems is that it is necessary to design our systems with an evolutionary mindset. A dynamic governance process is crucial. We need always keep in mind the spirit of the law is more important than the letter of the law. To maintain a decentralized network's integrity, we must firmly hold to our transcendental values as our eternal goal, instead of focusing on any formalizable set of rules of behavior.

Zero-sum games and Code-is-Law smart contracts

In the DAO hard fork, which split the Ethereum network as described above, the minority fork's principled objection was to maintain their ideal of Code is Law.

The Code is Law Principle holds that however a smart contract executes in following the logical steps of its program, that result is legal, regardless of the intent of the author of the smart contract. In the environment of Ethereum Classic minority, there is no such thing, legally, as a bug in a program. "The network does what the network does", is the whole of the law.

The Ethereum Classic fork consists of the members of the original Ethereum network who refused to adopt the new software that "fixed" the bug in the original DAO smart contract costing users in the early network more than US$50 million. The Code is Law Principle divided the community from those who were afraid the hard line would scare away investment because of the insecurity of living with every unintentional error.

Despite how alluring the simplicity of the Code-is-Law credo is, that extreme puritanical line is not an efficient solution for business.15 In the wider context of the future of business in a decentralized world, we must consider some practicalities. Unintended consequences of contracts arise in almost every business arrangement. We are not able to predict the weather one week in advance, much less the future of any complex human scenario. It is not strategically sound to make business decisions based on gambling. For mutually-beneficial, long-term cooperation to thrive, it is necessary to allow review of contracts assuming good faith from both parties when unintended consequences arise.

A zero-sum game is one in which the total rewards of the game are fixed beforehand and at the end of the game the rewards end up split between the players. If the economic system we construct in the Web3-enabled decentralized economy is merely

[15] But we are still sympathetic with the Ethereum Classic community. Especially if their motivation was to send a message to programmers and investors that messy thinking will be punished. Especially because a code is law environment is more efficient.

a zero-sum game, then cooperation is not possible as the only feasible long-term strategy is the pirate code: take anything not nailed down and leave nothing for your opponent. Then people are incentivized to play the most ruthless strategies available to hurt the other parties of a contract. Especially when anonymous parties are involved, this leads to an extremely degenerate situation. Adversaries are right to spend some extra energy to determine the minimum effort needed to satisfy the Code-is-Law smart contract.

However, the principle behind Code is Law is still an ideal to strive for. The efficiency and clarity that can be achieved in the long run by using the absolute mathematical logic and electronic speed of computer programmed and executed smart contracts is an indispensable opportunity for business. Relying on the smart contract regardless of outcome is extremely efficient. A Code-is-Law assumption is needed to deal with the unbelievably complex legal interactions governing the exponentially evolving business interactions that arise with AI-enabled IoT devices. How can we legally regulate the smart contracts which mediate between devices owned by many different companies and individuals interacting throughout the supply chain? New efficient processes lead to newer, more efficient processes. Business arrangements constantly adapt to these changing circumstances giving new contracts. The multiplicity of options in a dynamically changing market demands instantaneous legal enforcement without waiting for a centralized response of human interjection.

How do we choose between the efficiency of Code is Law and the business necessity of continuing an arrangement after the contract is technically broken? This is one of the many reasons there is not a robust decentralized economy using blockchain technology, yet. The decentralized solution still needs to be built.

1000 years ago, the Maghribis found their solution. And 200 years ago, modern democracies found theirs. A secure and meaningful reputational system and dynamic governance are two of the building blocks. We will explore contemporary applications of these ideas using recent advances in information technology in Chapters 6 and 7.

As smart contracts become more sophisticated, they may include many of the eventualities that most commonly happen in business situations. But as contracts become more complicated, that allows more complicated business arrangements. Which means more complicated unintended consequences and disputes. This is an eternal race as we develop more solutions that breed more problems. We can never design the ultimate program to solve all business problems or solve all human relations. Instead we need to wisely build a properly-incentivized, evolutionary environment for generating more efficient contracts which encourage better cooperation.

Application to PoS consensus algorithms

The goal of all existing proof of stake (PoS) algorithms is to create a protocol which incentivizes everyone in the blockchain network to behave the same way. We want everyone to run a single canonical program which eventually synchronizes everyone's perspective about the global state of the network, even though at any one time we each see only part of the state, because we share messages in the cloud by forwarding messages we receive from nearby nodes to other nodes near us. We want the protocol to make it very unprofitable for anyone to try to run a hacked program to gain any advantage—by doing less work for example (shirking), or ignoring messages that don't benefit us (censoring), or by sending false messages which break consensus (Byzantine faults), etc. PoS must punish anyone who violates the protocol strongly enough so that the vast majority of the network (at the very least 67%) will not copy their behavior.

Ethereum announced their intention to eventually implement a PoS algorithm since its inception in its original yellow paper. Despite the intense pressure they are under to perform, they have been looking in the wrong place, for years. A major component of their plan is to develop an algorithm that is "correct by construction (CBC)".16 That is, an algorithm that is rigorously provably correct: given a specific set of realistic assumptions about the nature of the network, they wish to prove that Byzantine behavior is impossible, or at least highly disincentivized. In the first place, this is simply false advertising. Most people don't understand that CBC doesn't mean it's mathematically proven to be perfectly resistant to all attacks; it's only resistant to the attacks the theorists consider reasonable at the time. Secondly, this is too ambitious to wait on for years as the network is wasting energy on PoW. Further, there is a better strategy that will be more secure and more efficient in the long run.

Finding a correct by construction algorithm that incorporates all possible, or even practical, assumptions of possible network statuses, is not possible. Given any set of assumptions about how the network will behave, you can always break those assumptions by valuing some other result.

Given the impossibility of creating an algorithm that will be perfectly secure in all circumstances, we should instead focus on developing a governance process that allows us to update our algorithm to adapt to the changing network circumstances to the security level required—an evolutionary algorithm that encourages improvements to the system with balanced rewards. Simply incentivize people to develop protocol improvements instead of unleashing attacks, by giving meaningful reputation.

[16] "Guide to Ethereum Proof of Stake and Casper" Online introductory resources: https://ethstaking.io/correct-by-construction-cbc-casper/ (Retrieved 8/8/20).

Similarly, any DAO will need to adapt to changing circumstances in their user base to maintain security and keep incentives aligned in service to their goals. We are always fated to engage in a technological arms race to maintain security in any realistic setting.

From the perspective of game theory, we think of a PoS consensus algorithm as a repeated game. The game is played each time a block is manufactured and accepted by the network. The goal is to design the game in such a way that the only profitable strategy is to follow the canonical protocol. Technically, we might think of PoS as a game design which makes the canonical protocol the unique subgame perfect Nash equilibrium.

This is the most challenging problem in decentralized computing, and the best minds in the area have been devoted to solving the problem since before Bitcoin was invented. Bitcoin's proof of work algorithm (PoW) is the first major implementation of a practical protocol to solve this problem. PoW has been successful enough to guarantee consensus for more than a decade in a network of millions of users, worth about 100 billion USD. Despite the fact that anyone can run any hacked version of the algorithm at any time from an anonymous account, there has not been a single protocol violation—meaning no message has been incorporated in the finalized blockchain which breaks the rules of the algorithm.

However, we suspect that the Bitcoin PoW algorithm is ultimately flawed. In fact, we believe that every consensus algorithm is flawed. Further we are confident it is not possible to create an algorithm that is not flawed. Impossibility results from mathematics abound. The vague goal of finding a perfect protocol for guaranteeing eternal universal consensus in the messy situation of real life, with constantly changing market environments, with arbitrary anonymous actors, is certainly too ambitious. The Folk Theorems of Game Theory display the vanity of that goal.

Application to reputation tokens

Game theory can prove several insights are valid with DAO design. To create the proper incentives these design choices should be considered.17 In this section we consider some traditional applications of game theory to explaining economics and the theory of the firm that DAO architects may consider in the future.

[17] See Avner Greif, "Institutions and the Path to the Modern Economy: Lessons from Medieval Trade", Cambridge University Press, 2006, p. 428--452.

First, we can disincentivize defection/betrayal by charging admission to the DAO. With the sunk cost of joining, it is more expensive to cheat, since then it would be expensive to rejoin even if members could be anonymous. This is described as costly signaling. It is more effective when the environment contains more cheaters, then people are incentivized to join the group that has differentiated themselves. (This can give some justification for the Denial of Service (DoS) fees that we use in the design given below.)

On the other side, blacklisting accounts can incentivize cooperation, especially with KYC18 protocols.

Another issue with repeated games to discuss is the end period. When is it better to choose the strategy of cooperating for long-term gain versus the motivation to choose short-term gain by betraying the opponent with defection? Short-term gain is the better choice if you are impatient, i.e., if the reward today is much more important than a reward tomorrow. In finance and game theory this is measured by the future discount factor. A guarantee of $10 in the future is rarely as valuable as $10 today. Natural economies have inflation because goods spoil in time, so it's better to invest your $10 today and get more than $10 in the future. Alternatively, you may be desperate for the $10 today and don't anticipate the need tomorrow. The fraction between the future and present value of the reward is the discount factor for each stage of the game. If the discount factor is too low, then it is better to defect. Or the dual problem is, if you know the game will end soon, then the future reward is too low, so you should again defect.

Therefore, the promise of future profits must outweigh the present value to ensure cooperation. The advantage of a large or global network in building a reputational system is that the reputation will have a more stable and predictable value. Though they are not as fungible are currency tokens, reputation tokens in P2P systems can be correlated with expected future salary, appraised (with hedging if they are tied to auditable past behavior), and sold. In this way, reputation tokens derived from smart contracts makes them more valuable and efficient than the vague notions of reputation from the past.

[18] KYC is the acronym for "Know Your Customer", which refers to identity verification protocols used especially in banking and insurance. KYC protocols are generally antithetical to the Web3 movement, which has a preoccupation with privacy and user control of their own information—especially because nothing is ever deleted on the internet: "Experts: Deleted online information never actually goes away", Chicago Tribune, August 21, 2015 https://www.chicagotribune.com/business/blue-sky/chi-deleted-online-information-never-goes-away-20150821-story.html (Retrieved 8/8/20).

Another advantage to digital tokens in open global networks is that the loss of opportunity from having your reputation slashed grows as the size of the network increases. Further, given the incomplete information due to anonymity, the value of the information from reputation tokens increases. When potential business partners have less knowledge of your identity, the knowledge from the number of reputation tokens you hold becomes more important. Moreover, the lack of personal knowledge encourages the members to devote more effort to fairly policing reputation tokens, so meritocracy is encouraged.

Next, a DAO design which allows anonymity must guard against various sockpuppet attacks. One strategy is to have one account which acts honestly and one which cheats. If the cheating account can funnel the gains to the honest account, without detection or punishment, this sets the system up for failure.

Another situation where DAOs with smart contracts have an advantage is with compliance. Algorithms can be written which exclude members who cheat from having access to their market. Punishment for cheating becomes automated and therefore credible. Free riding in policing can at least partly be eliminated by automation. In traditional business, members would police cheaters by withholding their business. But to make the threat credible, you would need to police the other members and punish them if they did business with cheaters. And so on, to make the threats credible, you would need to police those who did not police those who did not police. Algorithms can be written proactively to only supply contracts to those who have sufficient reputation. If your reputation is slashed, you will not be chosen by the algorithm.

Another motivation for transparency is that the value of reputation is directly related to how well punishment can be distributed in response to cheating. The more transparent the system, the more accurate and efficient policing can be.

How to build a DAO

"If, out of the present chaos, there is ever to come a world where free people live together peacefully, […] we shall have to furnish the pattern. It is not enough to restore people to an old and outworn pattern. People must be given the chance to see the possibilities of a new world and to work for it."
–Eleanor Roosevelt, The Eleanor Roosevelt Papers, My Day, December 16, 1941

Now we're ready to design a basic DAO. For specificity, we'll draft a DAO devoted to reviewing software for money. However, the basic design will apply to autonomously organizing any decentralized group for any purpose. In this case our group of software experts will be focused on judging new software products to determine how

safe or useful they are. The group will be called the Software Review DAO, or SRDAO.19

Fully describing how to organize a company from scratch is necessarily complicated and tedious. Describing how to organize a decentralized company, where members can be anonymous and geographically separated but still need to share power and profits, is necessarily extremely complicated and tedious. Readers should assume all design choices are options. More or less stringent requirements will be instituted in any specific instantiation of an actual DAO.

The SRDAO is a collaboration between anonymous software experts. They hash their reviews and post the hashes to the blockchain. The validated software can be hashed and recorded along with the reviews. Users can check whether the software they are downloading is correct if it has the same hash. The power of the review is checked by how many DAO experts staked their reputation to vote for it. Invalidated software will have pools where experts stake a lot of reputation to vote against it.

The fees will eventually come from the fact that your peer reviewer platform is valuable. The public would unconsciously use the SRDAO's work because their UI would only recommend software that had a sufficiently high review. Companies that want to prove their products are safe will eventually pay fees to have their software reviewed quickly (especially patches, e.g.).

Ultimately, we want to show how the members can be paid for their work. The challenge is to organize a group with no leader and no hierarchical structure. All members have the same roles. The members do only one job for one type of customer. (Different roles should have separate DAOs with separate types of reputation tokens. With interoperability being a main concern of developers with the goal of increasing network effects, separate DAOs will interact by subcontracting to cooperate.)

How do we set up the reward structure so that nobody can game the system? From the Folk Theorems, we know we cannot set up a perfect system. The best we can hope for, is to design a system which makes it easier to help the group than to hurt it. We need to set up a system so that the members themselves will be encouraged to police the bad actors, to protect their own profits.

For maximal applicability to general DAOs which have different goals, we will assume the worst-case scenario: Members from any location can join. Members could therefore be located in any jurisdiction and cannot be tracked or punished for any bad behavior by appealing to outside authorities—the only punishment available is to take

[19] The idea for the SRDAO comes from Clemens Cap and Benjamin Leiding, "Ensuring Resource Trust and Integrity in Web Browsers Using Blockchain Technology", Advanced Information Systems Engineering Workshops - CAiSE 2018 International Workshops, Tallinn, Estonia (2018).

away their potential future profits in the DAO. Members are all anonymous. Members might have multiple accounts (sockpuppet accounts). The only way to discourage malicious actors from joining is to charge money to join. Finally, we assume there are a significant number of bad actors who would harm the DAO if there were any opportunity to do so, whether or not it were profitable. Therefore, a nominal fee to join will be charged as resistance to basic DoS attacks.20 This fee merely needs to be high enough so that the effort to police the bad actors is profitable, but not so high that it prevents people of good will from joining. This number depends on the market environment. Automation should make this feasible.

The only levers of power the DAO can wield over their members is to reward them by sharing DAO profits or punish them by withholding those profits. The DAO will therefore institute a reputation token system to push these profits as far as possible into the future, to give time to review the members' actions and encourage good behavior. For specificity, we assume the reputation tokens will be created and tracked using a blockchain such as Ethereum (see Section 2.7, above). Alternative P2P approaches are available, such as a distributed hash table, which would be more efficient (especially for voting) and less secure.

A periodic reputation-weighted salary will distribute all fees the DAO earns to all members. Individuals who perform tasks that bring fees to the DAO will be rewarded with reputation tokens, not the fees. Members who own more reputation tokens share in a larger percentage of the fees. This solves the sockpuppet attack: if you have 10 accounts with 1 reputation token each, it's the same as 1 account with 10 reputation tokens.

So, there are two types of tokens to keep track of, reputation tokens and fees. The fees come entirely from customers who engage the SRDAO to review their software. These fees are fungible currency such as bitcoin or USD.

The basic function of the SRDAO is as follows:

A customer uses the SRDAO smart contract (SC) to engage a reviewer by encumbering the fee in the SC and uploading the software to be reviewed.

The SC randomly picks a reviewer/member from those members available. (The only other smart contract in use will be an availability smart contract which members engage by encumbering their reputation tokens.)

The reviewer evaluates the software according to the principles the SRDAO has previously collectively agreed upon.

The reviewer posts their review.

[20] Denial of Service (DoS) attacks happen when anonymous adversaries flood the system with automated requests for superfluous tasks, thus preventing the network from engaging in productive work.

The review triggers a validation pool—a voting pool where any member can stake their reputation by voting to approve or disapprove of the review.

The SC mints new reputation tokens in proportion to the size of the fee.

The new reputation tokens are staked half in favor of the review in the reviewer's name; half are staked against and left unassigned.

Majority wins and the reputation tokens are split amongst the winners.

Results are posted displaying how the software was received by the reviewer and whether the SRDAO community agreed.

The fee is split amongst the entire SRDAO in proportion to their reputation holdings (reputation-weighted salary).

We are assuming that every stage of this process is automated. The most energetically intensive step is 4, where the reviewer reviews the software, but even this is imagined to be largely automated once the SRDAO is fully operational. The idea is that members use a uniform preapproved protocol for doing everything. That way, participating in the validation pool is also automated.

In a well-functioning DAO every validation pool will result in nearly unanimous votes—the only contrary votes should be those shirking their duty, who will lose their reputation tokens for coming to the wrong conclusion. The validation pool is simply for policing the group and maintaining unity. It is not for gambling.

Reviewers don't get fees directly, they get more reputation tokens, if they are successful in the validation pool, because of Step 5b. The rest of the members share newly minted reputation tokens for participating in the policing of the DAO.

In the beginning, this Software Review DAO will review software for free. In this beginning stage the SRDAO is building value, until the collection of reviews becomes useful.21 The reviewers/members will gain reputation tokens for their work. Once the SRDAO attracts fees, members' early seemingly-altruistic efforts pay off with the reputation-weighted salaries. Since all fees are shared with all members, this eliminates the incentive to positively review software just because a company sends a big fee. Basically, a single expert reviewer cannot be bribed as easily if the fees are shared with everyone.

The creation of every step of the DAO from the very beginning follows this same 7 step procedure.

New member review. A new member submits an application smart contract as if they were a customer. The fee would go to the DAO, new reputation tokens would be

[21] Insights from past open source software projects are essential for initiating effective DAOs. A useful guide to the values that encourage the success of these groups is given by Katherine J. Stewart and Sanjay Gosain, "The Impact of Ideology on Effectiveness in Open Source Software Development Teams", MIS Quarterly, Vol. 30, No. 2, pp. 291-314, (2006)

minted and staked in the applicant's name in Step 5b. The rest of the DAO members would validate the application, or not, to police the DAO. The first member of the DAO would follow the same procedure to receive the first newly minted reputation, since there would be no one to vote against them.

The second member would need to convince the first member to vote for them, because the new tokens are staked half in favor, half against—so the first member would have complete control of whether or not the validation pool resolved in their favor.

Proposals for new protocols. Suggestions for updating how the DAO should run can be submitted for review by staking reputation as a fee. The process would mint new reputation tokens, and if the protocols are accepted, the proposer would gain half of them, the rest would be shared with the group for participating in policing.

The system is evolutionary in structure. The SRDAO continually rewards work on new software reviews and it encourages the creation of more sophisticated protocols for how to run the DAO. They should continually create more sophisticated protocols for reviewing the reviewers' reviews.

The idea is that for most software reviews, a reviewer does several standard statistical tests, explaining which ones were done in their review. The SRDAO will have developed a common reviewing program, which if run faithfully, will always give the same result. Then when the other experts validate the review, they just have to run the tests themselves and upvote the reviewer. The idea is that a faithful reviewer who is following SRDAO protocol will win every single validation pool, and never lose a single reputation token. Those who shirk will eventually be found out, and lose their reputation.

How does a reviewer review a new piece of software that requires new techniques that aren't spelled out in canonical SRDAO protocol precedent? Propose a new protocol.

How does the DAO set the fees? How does the DAO decide how much to reward protocol creation compared with day-to-day work of reviewing a standard software product? Much more complicated procedures can be developed, since smart contracts are technically capable of any type of business logic. We will explore more options for DAO governance in Chapter 7 when we explore how different choices of reward and punishment rules (particularly at Step 5 a, b, and c) encourage behavior to reflect different values at different points in the life cycle of a company.

However, we mostly rely on the market for many of our answers. This is not a cop out. The market is well understood to be the most efficient mechanism for price discovery. Secondarily the market determines how many members the DAO can support in its network, and how much work they should do. The market's answers find the right equilibrium between all the industries and companies to determine how much

of each work and good is required to keep civilization running efficiently. Decentralization improves the market in its efficiency at price discovery and liquidity.

The liquidity of a market is its momentum. Liquidity is measured by its velocity (dynamism, motion, rate of transactions), but also by its mass (how valuable are the total goods or assets). When there is too much mass in a market and not enough velocity, it can gunk up the engine. When there is too little mass in the market, it can be cleared too quickly, resulting in vapor lock. One of the problems with centralization is that the power accumulates, which is bad for market efficiency, for liquidity. Monopolies are a threat to market liquidity as they can have too much mass or too much velocity. Decentralization is better, as it means the mass and velocity are uncorrelated meaning more liquidity and therefore more market efficiency.

This exact process of Steps 1-7 can be cloned to build a news story review DAO. Reviewers could judge stories' veracity according to the standards the DAO chooses. As before, reputation in the Review DAO becomes valuable once News DApps steer viewership to stories based on your DAO's reviews. Then media sites will pay fees to get their stories reviewed more quickly. Reviewers will be incentivized to give honest reviews despite fees, because the reputational system rewards members mostly based on future fees. As members have all power of policing, they are naturally incentivized to encourage reviewers to maintain the integrity of the system.

Notably, the transparency and openly reviewable nature of a DAO greatly adds to the trustworthiness of the system. Being eternally open to audits and reviews from anyone on the planet greatly improves the integrity of the system. Such improvements to transparency and accountability in the institutions of media and education have the opportunity for improving many aspects of society.

Consider also how this process can be cloned with little variation to initiate a decentralized organization devoted to most any goal.

We'll discuss further details of reputation token creation and governance in Chapters 6 and 7.

Bibliography

Formating example:

Evans, Dave (Apr. 2011). The Internet of Things: How the Next Evolution of the Internet is Changing Everything. CISCO White Paper, https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf (accessed June 1, 2020).

Wikipedia. Last Universal Common Ancestor, https://en.wikipedia.org/wiki/Last_universal_common_ancestor (accessed June 1, 2020).