Simple Tokenomics for a Proof-of-Stake Utility Token

Noam Nisan (StarkWare and Hebrew University)

This document discusses “Web3 platforms with proof-of-stake and a utility token”, a family of platforms that is quite common in the “blockchain world”. We readily unpack what each of these terms means, and Table 1 presents the largest platforms of this type as of early August 2023 with some of their significant “financial” indicators[1].

Platform	Token	Market Cap ($B)	Staking Ratio (%)	Yearly Reward (%)	Adjusted Reward[2] (%)
Ethereum	ETH	224.3	18.7	5.2	5.1
Binance	BNB	37.2	14.8	2.1	8.0
Cardano	ADA	10.2	62.3	3.1	0.4
Solana	SOL	9.4	70.6	7.1	-0.3
Polkadot	DOT	6.3	44.9	14.6	7.1
Polygon	MATIC	6.2	38.8	5.3	2.9
Tron	TRX	5.1	47.7	3.5	9.8
Avalanche	AVAX	4.4	62.4	7.4	1.2
Cosmos	ATOM	3.0	70.1	20.4	2.8

Table 1: Proof of Stake systems with utility tokens having at least $3B marketcap
(Data retrieved from stakingrewards.com on August 7th, 2023)

Our aim here is to suggest simple and broad principles for thinking about the economic aspects of such systems and of their tokens, to be called their

“tokenomics”. We aim for extreme simplicity and any specific system will naturally require more complex analysis according to its specific goals, constraints, and environment. Still we hope that the point of view presented here can serve as a useful way to start thinking about the issues as well, perhaps, as guidelines for the first approximation of design.

1. The Type of Systems that we are Discussing

In which we unpack what we mean by “Web3 platforms with proof-of-stake and a utility token’’ and argue that such platforms must offer some utility to users, must grow to a sufficiently large size, and must reward their operators.

1.1. Web3 Platforms

We use the term “Web3 platform’’ to refer to any computational platform that provides some online service in a way that achieves common agreement and trust without reliance on any trusted central party. Basic examples are cryptocurrencies such as Bitcoin, digital economy platforms such as Ethereum, various decentralized “L2‘’ layers that add value to Ethereum, or specific decentralized application platforms for finance (DeFi). The point of these systems is that they should continue to operate in a way that you can trust without relying on the proper behavior or even existence of any single company, institute, or government. Basically the single trusted party is replaced by a consensus of a large number of small parties.
One may certainly question the desirability or importance of such systems that sidestep conventional battle-tested mechanisms such as banking and finance, but this manuscript will take it as given that many people desire such systems and view the lack of reliance on a central party to be desirable and significant for certain applications.

The level of trust that a Web3 platform provides is clearly a function of the size and quality of the large set of parties that cooperate to underpin the trust in the system. It follows that such platforms have significant positive-feedback network effects: the more the platform grows, the higher the trust in it, thus the higher the value that it provides, attracting more participation, and thus increasing growth even further[3].

A critical requirement from any Web3 system is to initially grow and then to maintain a size that delivers significant network effects.

1.2. Proof of Stake

As the security of a Web3 system is based on the cooperation and agreement of many small parties, a key challenge that every Web3 system must address is that of “sybil-resistance”: how can we be assured that what appears to be a large set of parties is not really just a single party masquerading as many. Following the Bitcoin system, early systems addressed this challenge using a “Proof of Work” mechanism where the parties underpinning the security of the system needed to exhibit computational power. As Bitcoin grew in popularity, the amount of this computational power grew to the point that the required electricity power rose to be a significant fraction of world electricity use and has non-negligible effects on global warming.

While there are a few suggestions for other types of sybil-resistance, such as “proof-of-humanity”, i.e. the identification of actual humans, it is probably fair to say that currently the only other alternative with significant use is “Proof of Stake”. In such systems the participating parties must own certain system “tokens”, and the amount of tokens that they hold is what gives them their “identity” in the system. Specifically, agreement in the system is defined as what is agreed to by parties who collectively hold a majority (or maybe more than a majority) of the participating stake.

There is a diverse literature on proof-of-stake vs. proof-of-work systems, but here is the economic perspective of how things work in a typical proof of stake system. Initially, the platform “mints” some amount of tokens and allocates them in some way. To participate in running the platform, an operator must acquire tokens, in some market for tokens that will emerge, and “stake” them, ie. lock them in the platform, where they serve as a collateral for his proper operation in the system. In return for staking the tokens and for the ongoing work of participation in the operation of the platform, the stakers are usually rewarded by the platform, getting more tokens (which they can then sell in the open market.) Depending on the platform’s protocol, these rewards can either come from fees paid by users of the platform or they can be newly minted tokens. If the rewards come from new minting then clearly the total token supply increases (i.e the token is inflationary[4]). An alternative possibility for rewarding the operators is to give them the power to extract some value from the users of the system, what is often called Miner Extracted Value (MEV).

Stakers in proof of stake platforms must be rewarded either from user fees, or from minting of new tokens, or by extracting value from the users, or by some combination of these.

In the table above we see "tokenomic" data for the largest staking platforms including Ethereum valued at over $200B and eight other multi-billion dollar platforms. (Many smaller platforms exist with about 50 of them valued at over $100M each as of the time of writing.) As we can see, the nominal rewards provided to stakers (yearly, as a percent of their staking amount) vary in the range 2%-20% with a median somewhat over 5%. Adjusting the rewards by the token-inflation amount, the real rewards vary in the range of 0%-10% with the median at about 3%. Not all of the tokens in these systems are staked and the fraction staked is in the range of 15%-70% with the median close to 50%[5]. One of the goals of this paper is to suggest a principled way to think about these numbers.

1.3. Utility Tokens

There are many types of tokens and there are multiple ways of classifying them. For this paper we are interested in the classification by economic purpose. This classification looks at three types of tokens: payment tokens, utility tokens, and security tokens. Payment tokens are meant to serve as "money" typically in the sense of being a medium of exchange and a store of value. Typical examples are Bitcoin on one hand and the many stable coins on the other[6]. Security tokens are financial instruments that provide their holders with certain legal rights or claims against an issuer as do financial securities like stocks or bonds.

Utility tokens can be used to automatically[7] get some service from the platform, allowing a user to obtain some utility from it. Most typically utility tokens can be used to pay fees for using the platform, where the platform provides some service to such users. As a typical example, the Ethereum blockchain provides the service of running transactions on the Ethereum "Computer in the sky" public ledger, a service that is desired by many users who are willing to pay significant money for it. The native token of Ethereum, ETH, is the only way to pay for this service so naturally potential users of the Ethereum blockchain must buy some ETH tokens from some willing seller and then pay the Ethereum blockchain with these tokens.

When one takes a purely “utility”-based analysis of a Web3 platform, it becomes clear that a key goal of such a system is to indeed provide as much as possible utility to its users. Naturally, providing utility in a Web3 platform will require maintaining sufficient trust and openness as well as satisfying other platform-specific requirements. Having a token turns out to be a key ingredient in enabling the required trustless collaboration and taking this utility-providing view of our platform, the token’s purpose and hence its tokenomics should serve this goal of providing utility as well. This type of clean “micro-economic”-founded analysis of the token is what we will undertake here.

Clearly most utility tokens may have additional functions and may serve to some extent as payment tokens, for example. One may well suspect that the current value of ETH is not only a function of its utility for running transactions on the Ethereum blockchain but also of its use as a store of value and means of payment just like Bitcoin is. Our analysis will be appropriate when most of the value of the token, or at least a significant part of it, is derived from its utility-token aspects.

2. Micro-Tokenomics: Fees and Social Welfare

In which we describe the micro-tokenomics of the token, focusing on the transaction fees that users need to pay in order to use the platform. We argue that the optimal transaction fee is the marginal cost that the platform incurs for running the transaction, including congestion costs if there is congestion. As a platform with a utility token provides, by definition, some service to its users, a type of market for this service must emerge, a market that will govern who gets the service and how much they pay for it. This section provides the basic analysis of such a market.

In keeping with our aim of simplicity we leave our discussion as simple as possible while still capturing, we believe, the essential economic features of utility-based Web3 systems. In particular we stick to a “static” analysis and avoid issues of timing and dynamics which are often more difficult to handle, but we believe should be approached using the same principles as the static case.

2.1. Platform Goals and Social Welfare

Our first order of business is to figure out what the platform should attempt to optimize. While the knee-jerk response would be “make the builders of the platform rich”, such cynicism ignores the behavior of the intended participants in the platform’s ecosystem and does not suggest any prescription for making decisions. We defend the exact opposite point of view: the goal of the platform is to maximize the total value that the platform brings to “the world”, what economists sometimes call maximizing social welfare.

Let us start with the normative point of view: what *should* the platform optimize for? If one views the platform as a corporation and the platform’s tokens as its stocks, then one would naturally attempt to optimize the revenue that the “stockholders” receive. This point of view is not aligned with the Web3 community which does not like to view its infrastructure as a corporation but more as providing a public service to its *users*. The Ethereum blockchain is a good example where the owners of ETH do not directly make any profits from the operation of the Ethereum blockchain unless they stake their tokens. Going back to our distinction between a security token and a utility token, the former aligns well with maximizing revenue for token (stock) holders while the latter—what we are focusing on—aligns with maximizing value for the platform ecosystem as a whole, including (mostly) its users.

If the normative discussion above seems too naive or pretentious, let us also consider a more practical point of view. Suppose that some participants have other less philanthropic goals such as maximizing their private revenue as token holders. How may they do so in the long run? Due to the network effects that are the inherent drivers of any Web3 system, the most important factor for a platform would be growth. A platform that grows more will survive and not only bring more “social welfare” to its whole ecosystem but also to creators and token holders. The main way one gets growth is by making sure that the platform indeed supplies as much utility as possible. Not only will this attract users to the platform due to the direct value that they get, but also “optimizing for the users” provides a better public message that is important in the Web3 community. The appropriate metaphor for this model of platform goals may be more like a nation state than a corporation: the goal is not to grow shareholder value at the expense of any other good; the goal is to grow an entire economy that will ultimately improve the lot of all participants[8]. Translating this to the day to day operations of the platform we end again with the working goal of maximizing social welfare.

The working goal of a Web3 platform with a utility token should be to maximize the social welfare that it delivers.

2.2. How Can We Maximize Social Welfare?

So, suppose that we indeed want to optimize social welfare either for normative or practical reasons, how should we go about it? To begin with, clearly the platform must provide some useful service, so for the rest of the discussion we assume that it does. Let us be a bit more specific, taking us into an economic model. When we say that it does something useful, it has to be useful for *someone*. We will call these "someones"—the people who can get value from the platform—the (potential) ***users***. Let us abstractly call a unit of "service" supplied by the platform a ***transaction***. Operation of the platform will likely require some resources and effort, and let us call the people (or corporations) who provide these the ***operators***.

At this level of modeling, the basic question of social welfare maximization boils down to *which transactions should be serviced by the platform*. There may be two reasons why we should *not* service a transaction even though some user gets value out of it. First, it may be that the costs (effort and resources) associated with serving the transaction are higher than the value that it gives to the user, in which case, servicing the transaction gives a total negative benefit. Second, the platform may likely have some limits on its capacity[9], and if there is more demand for transactions than it can supply, it will have to choose the most "valuable" ones and ignore the others. In order to proceed in our analysis it will be useful to dive into a very simple economic model of such scenarios.

2.3. The Basic Economic Model

So let us try to describe the basic economic model that captures the essence of our situation: There are multiple transactions $i=1,2,\ldots,N$, that want to be serviced by the platform. Each transaction i has a user that initiated it and that user has an associated value v_i for it. A transaction also has an associated marginal cost c_i that the platform (via its operators) must incur in order to service it (on top of the other transactions that it is already serving). While in classic economic theory the marginal cost of a unit of a good is often a function of how many other units are already produced (increasing or decreasing marginal costs), in our situation it is probably safe to consider the cost of a transaction to be fixed (after some fixed cost for even starting to operate and until a point where we have reached some capacity limit of the platform). Maximizing social welfare means choosing that set of S of serviced transactions

that maximizes $\sum_{i \in S} (v_i - c_i)$ over all possible sets of transactions that fit within the capacity of the platform.

So, in this model, which transactions should we service? Well, if we are not at the platform's capacity limit, then we should service any transaction with a positive value of $(v_i - c_i)$ i.e. that has $v_i > c_i$. How can this be done? While we may assume that the platform can compute (or at least estimate) the cost c_i associated with servicing a transaction, the value v_i of a transaction is subjective in the eyes of the user that is interested in that transaction and thus only known to him. So here is the basic economic trick of doing so: charge the user a transaction fee equal to c_i for servicing his transaction. In such a case the user will choose to run his transaction if and only if his private value v_i is higher than the cost, $v_i > c_i$. This is called "marginal cost" pricing and is a basic fact presented in a "econ 101" course: in order to maximize social welfare the price of a unit should be equal to the "marginal cost" of supplying a unit[10].

In order to optimize social welfare, transaction fees should be set to their marginal costs. This aligns the user's net utility with social welfare.

In the case of congestion, the associated "marginal cost" should also take into account the effect that servicing our transaction has on other transactions which did not get serviced because of our transaction. In this case the fee for a transaction should take into account not only the direct cost c_i of the transaction but also the "congestion costs": the net loss of social welfare that it caused to the other users. Let us see how this works out in the simplest and most common case.

Single dimensional gas model: Here is the simplest and most common model for describing the capacity constraints of a Web3 system. Each transaction has some size *si* that describes how much of the system's resources it uses (borrowing Ethereum terminology, this may be called the "gas" used by the transaction), and the system has some capacity *K* of total resources (i.e. of gas). Thus a set *S* of transactions is feasible if $\sum_{i \in S} s_i \leq K$ *and maximizing social welfare means maximizing* $\sum_{i \in S}(v_i - c_i)$ *subject to this constraint. Furthermore in this model the cost associated with running a transaction is considered to be proportional to its size* $c_i = \alpha s_i$*, (where α is some global constant). While in general there is no efficient algorithm for this optimization problem (as it is the classical Knapsack problem), there is a well known greedy approximation algorithm: sort the transactions according to decreasing value of* v_i/s_i *and service transactions from the top until a point where taking the next transaction will exceed the capacity limits (or until* $v_i < c_i$*). The "price of gas" will be set to* $g = v_i^*/s_i^*$ *of the last accepted*[11]

transaction i^* or, in the case of no congestion, to a minimum price $g=\alpha$, and the fees will be proportional to this gas price $f_i=gs_i$[12].
The model just discussed above is very simple and naturally ignores many aspects found in real platforms. Nevertheless, the main economic lesson of our simple model should continue to hold very generally: in order to maximize social welfare we should charge transaction fees that are equal to marginal costs. When there is congestion then transaction fees should also include "congestion costs".

2.4. Transaction Fee Mechanisms

While we have identified the required fees that will ensure social welfare maximization, we would need to also define a concrete mechanism that will allow our platform to actually charge these fees. Such mechanisms must take into account that both users and operators of the platform act rationally, "strategically", each attempting to optimize his own utility, and furthermore that collusion between an operator and multiple users, whether real ones or fake ones, is possible. While users should always be assumed to act strategically, we only need to worry about the strategic behavior of operators whenever they have some leeway in their behavior in the sense that other operators cannot "catch" them acting not in accordance with the prescribed protocol. Operators with no such leeway need only be incentivized to keep participating via means of some bulk payment, "block reward" for continued participation. When leeway exists, e.g. when an operator decides which transactions to accept, the platform protocols need to ensure that the operator is incentivized to act as desired. Perhaps surprisingly, even simple mechanisms can obtain the required fees as their equilibrium.

Pay your bid mechanism: Take for example the simplest type of mechanism, "pay your bid", à la Bitcoin, for deciding which transactions to accept, and let us see why we expect it to approximately charge marginal costs (including congestion costs) and hence approximately optimize social welfare. The basic mechanism works as follows: for a particular point in time—block—there is a single operator (miner) that gets to decide which transactions get into it. For our purposes it does not matter how this operator is chosen, just that one is chosen and that the protocol ensures that his decision will, most likely, become consensus. Users make bids for their transactions, and the chosen operator can accept any subset that he desires of these bids—within some given capacity—

and charge the bid for any accepted transaction. So what do we expect to happen in the long term, at equilibrium? If we view our situation as an economic market (for space for transactions) one would hope that the market reaches equilibrium, and in such an equilibrium the fees to be equal to the marginal costs, and social welfare to be maximized.

Equilibrium in the gas model: Let us return to our model of a single-dimensional resource where each transaction i has value v_i, size s_i, and cost is proportional to its size $c_i=\alpha s_i$, and the total capacity of the block is limited by K. Now each owner of a transaction makes a bid b_i. Looking back to the operator that gets to decide which transactions are accepted, it is clear that an operator that is paid the bids b_i will accept the set of bids S that maximizes $\sum_{i\in S} b_i$ which (ignoring integrality constraints) means taking the set of bids with highest ratio of b_i/s_i until the block capacity is reached. We expect the bidding dynamics to allow bidders to find and bid (in the long term, approximately) the lowest value of b_i that will make their transaction be accepted (as long as $b_i \le v_i$ as otherwise they are not willing to pay b_i). The equilibrium reached under this assumption will have bidders whose value per size, b_i/s_i, is high enough bidding at the "equilibrium gas price" p^*, where bidders with lower value will bid less than that. I.e. each bidder with $v_i \ge p^* s_i$ will bid $b_i=p^* s_i$ while bidders with $v_i<p^* s_i$ will bid a lower value, say, $b_i=v_i$, and where the total size of high bidders exactly fills the block capacity (as otherwise the operator will take an additional transaction with a lower value of b_i/s_i leading the other users to discover that they can reduce their bid and still be accepted.) This maximizes social welfare, $\sum_{i\in S} (v_i - c_i)$ as long as $p^*>\alpha$ (recall that $c_i=\alpha s_i$) thus leading to welfare maximization. To handle the non-congested case where transactions with $v_i \ge c_i=\alpha s_i$ do not fill the block's capacity, the system must mandate a minimum gas price $p^* \ge \alpha$ as part of the protocol[13].

Incentive-compatible mechanisms: One may worry how quickly and to what extent the pay-your-bid reaches (at least approximately) this equilibrium and how the users will figure out the magic parmeter p^* which they need in order to bid appropriately. More sophisticated fee mechanisms like EIP-1559 used in Ethereum can make the bidding process more transparent ("incentive compatible" in the jargon of mechanism design) and hence directly lead the system towards an efficient equilibrium, which will maximize social welfare and have the fees be equal to marginal costs[14]. Significant knowledge exists on how to design such mechanisms[15] and the existing knowledge generalizes to more complex and realistic scenarios.

Extracting value from users (MEV): It is probably prudent at this point to consider "implicit fees" taken from users when the mechanism allows operators to extract some value from user transactions. The nature of such extraction will certainly depend on the platform's services, but a typical example in blockchains is where a validator that creates a block can add transactions of their own that "frontrun" certain user transactions, hence transferring value to themselves. This opportunistic extraction of value is unrelated to transaction serving costs or to congestion and thus these implicit MEV fees are not aligned with the economic well-functioning of the system and, for example, may drive away users whose transactions can be taken advantage of. It follows that mechanisms should minimize the possibilities for such extraction, even though it may sometimes be impossible to eliminate them completely.

2.5. Bottom Line: Micro-tokenomics

So the bottom line of this section is that Web3 systems with a utility token should aim to maximize social welfare (aka value added) that they offer. This will happen when the fees charged are equal to the marginal costs of the transactions served (including congestion costs). Indeed there are economic mechanisms that can lead to this happening. While the details may be complex depending on the complexity of the platform, the basic principles should still hold.

3. Macro-Tokenomics: Staking Costs and New Minting

In which we describe the macro-tokenomics, focusing on the relations between the rate of minting new tokens, the rewards to stakers, and the security obtained from staking. Our main argument is that staking rewards should cover the capital costs of stakers, should best be paid from new minting, and should be the main factor determining the minting rate.

The previous section focused on transaction fees, what may be considered to be the *micro*economics of a Web3 platform that provides some utility to its users. We now switch our attention to *macro*economics of the platform: how is the overall system funded and how is the token managed. Again we emphasize our focus on simplicity and generality, noting that realistic systems will likely have more complex considerations, but hoping our analysis can still serve as a useful departure point. We start with what we consider to be the main gap between our microeconomic analysis above and what is needed for a system to be economically viable.

3.1. Fixed Costs

The formula of charging transactions according to marginal costs sweeps under the rug a central question, that of "non-marginal" costs. Let us be more precise. In all our discussion above, the only cost that mattered regarding the servicing of some transaction *i* is the additional—marginal—cost of how much more would servicing this transaction cost on top of the cost for all other transactions. This is really what determines whether we should service this extra transaction (assuming that we already decided which other transactions to serve). Consider an example a case where the cost for serving *N* transactions is given *$100+N*$1*, i.e. serving 9 transactions costs $109 and serving 10 transactions costs $110. In this case we would say that there is a fixed cost of $100 and a marginal cost of $1. Despite the $1 marginal cost, the *average* cost for servicing the 10 transactions is $11. If we only charge the marginal costs then we can only charge $1 from each transaction, but where would the missing $100—the fixed costs—come from? This problem of running a deficit when charging only marginal costs manifests itself whenever the marginal costs are smaller than the average costs, which seems to be the typical case in blockchains[16].
While in the "real world" any fixed costs must ultimately be paid by the users themselves, hence making marginal-cost pricing infeasible[17], in a tokenized platform it is possible to pay for the fixed costs from minting new tokens. This has the advantage of keeping the fees at the level of marginal costs. Ofcourse, someone will still be paying for the fixed costs and that someone is the aggregate of all token holders. I.e., minting new tokens means that we inflate the token in question, presumably reducing its value, and hence each token holder is effectively losing a small fraction of the value of his tokens.

One may wonder to what extent it is justified or desirable to put this burden on the token holders. We argue that this is the least-bad alternative. First, it allows us to charge the users only marginal costs hence maximizing use of the system, which as we argued above is our basic goal. Second, once the use of the system is indeed maximized, one would expect the total value of the platform to increase, hence increasing the value of the token, which will compensate the same token holders that have paid for the fixed costs.

In order to allow marginal-cost pricing, any fixed costs associated with operating the platform should best be paid by minting new tokens.

Looking at existing blockchains, this is currently more or less the norm: new tokens are minted to pay for "block rewards" that reward miners, stakers, or sequencers independently of the specific transactions in their block. These block rewards essentially cover the associated fixed costs, while the transaction fees are an additional component of the payment to the operators.

The suspicious reader may wonder about the sustainability of this process: does it make sense for the token holders to keep subsidizing the platform indefinitely? What is the "end game" here? There are several possible such "end games": first there is a possibility of sustainable growth of the platform as part of the sustainable growth of the world economy[18]. Another possibility is if the growth in demand for the platform's services outpaces the growth in the platform's supply. In this case, the congestion fees will keep growing (as they should for the micro-tokenomic reasons detailed above) and since these are not actual costs borne by the operators of the system, congestion fees can cover the fixed costs[19]. Yet another possibility is that as the use of the platform grows, the fixed costs become small compared to the growing marginal costs to a point that they can be added as a small overhead to the fees without causing significant distortion. Finally, if none of these suffice, one may imagine that in the long term, *after sufficient growth has occurred*, then indeed the platform will gradually increase the fees beyond the prescribed marginal costs as is done in the "real world".

3.2. Staking Costs

The most significant fixed costs in proof of stake systems are often the financial costs of staking, sometimes called "security costs" or "capital costs" or "opportunity costs". Specifically, a staker that holds a certain amount of tokens and stakes them in the system is forgoing other uses of this capital, either directly or other uses of fiat currency (e.g. US$) that he can get for selling the tokens. These costs are basically financial costs whose magnitude is determined both by the external financial environment (e.g. the current interest rate), by platform-specific aspects relating to the real and perceived risks of staking the token, and by other possible uses of the token (e.g. providing liquidity in AMMs). Specifically, if the total stake is *S* (measured, say, in US$), and stakers get an annual return of *r%*, then the total annual cost of staking is *r%*S.*

Ethereum example: Let us compute as an example the staking costs of the largest staking platform, the Ethereum network, according to the early August

2023 data shown in Table 1. About 19% of the tokens of the ethereum platform were staked at the time, and since the total value of all tokens was $224B, the total value of staked tokens was $42B. According to the table, these tokens were getting a yearly reward of about 5% so the total annualized staking costs were over $2B. The number of transactions on Ethereum is around 400M/year (slightly over 12 transactions per second) so we are talking about over $5/transaction which is likely the lion's share of the total costs of operating the Ethereum platform.

3.3. Staking Rewards

While the platform will need to mint new tokens to cover operators' costs, it is naturally best to only mint as much as is required since new minting is a burden on token holders. The protocol needs to ensure that the rewards allocated incentivize the operators to participate and act properly and will thus have to find the right balance between sufficiently rewarding operators and minimizing new minting. We claim that the incentives can usually be quite easily handled at a micro-economic level, and thus the main factor that should determine the minting rate is that of rewarding operators for their capital costs.

As an example, let us go back to the "standard" model of a blockchain where for every block there is a single operator, the leader, that builds the block and incurs most costs for doing so, while other operators essentially only "sign-off" on the block using some consensus protocol. For proper operation of such a system, the leader must be significantly compensated for his effort in building the block, and all other operators must be compensated for continuously "stayin' alive". The latter is usually easy to do since they do not have significant agency, so there are no significant incentive constraints from a macroeconomic point of view[20]. Rewarding the leader of a block is typically a more delicate issue since the leader has very wide discretion so we need to incentivize him "just right". This, however, is directly handled by the transaction fee mechanism that exactly motivates the inclusion of the right transactions using the correct marginal-costs pricing in a way that takes the leader's incentives into account as well. Beyond these incentives that properly handle marginal costs, we only need to compensate the leader for his fixed costs, which large enough fixed "block rewards" will do.

The main factor that should determine the minting rate is that of rewarding operators for their capital costs.

In the rest of our analysis we will take the reward rate that operators require from the platform to be determined by the stakers themselves in ways that depend on their own financial calculations. The stakers' financial considerations may be related to the financial environment, to the perceived risk and potential of the platform and of the token, and to alternative uses of the token. While it may be possible to apply various financial models to estimate how the required reward rate depends on other financial parameters (such as the outside interest rate or the historic volatility of the token's exchange rate[21]) we will not be needing such estimates and will continue to take the required reward rate as a given. Empirically, on the different large platforms listed in table 1, stakers' annual rewards were between 2% and 20%, with typical rates in the range of 3%-7.5% and a median of around 5%. This significant variability likely depends on a host of factors and undoubtedly there is also significant "noise" regarding the exact meaning of these numbers. Nevertheless we can get a pretty coherent picture of reasonable reward rates, putting them in the same ballpark as typical bond or stock yields.

3.4. New Minting

A platform must provide sufficient staking rewards to its operators, or otherwise operators will not agree to participate in operating the platform. As described above, what is "sufficient" is determined by the stakers themselves and is mostly a financial question. So how can a platform design for this? This question is what we address now. Our basic analysis will identify the fixed costs with the staking costs and will assume that the source of staking rewards will be newly minted tokens which on the average will be split pro-rata between the stakers. We will only focus on the overall quantities of minting and of rewards and not on their exact composition, which we assume can be properly handled by the transaction fee mechanism.

To proceed with a simple analysis, let us focus on the following parameters: (1) the annual rate of minting as a fraction of the total existing token supply, (2) the annual rate of staking rewards as a percent of the staked amount (3) the staking rate which is the is amount of staked tokens as a fraction of all outstanding tokens. The overall equality that governs this relation and hence the macro-tokenomics of the platform is:

(New minting in %/year) = (Staking Rewards in %/year) * (Staking Rate)

We already discussed the staking reward rate, so let us now look closer at the new minting rate. This rate is determined by the protocol which should specify when new tokens are minted (or, conversely, burned). Under our assumption that new minting is what pays for staking rewards, it is equal to the net sum of all annual rewards allocated by the protocol[22]. Specifically, in a block-based protocol, if the protocol specifies a total reward of R tokens per block (for all operators combined, net any burning, on average), where the total existing number of tokens is S and there are N blocks per year then the annualized rate of new minting is RN/S.
One may note that the staking rewards under this equation are "nominal" i.e. do not take into account any inflation in the token's value. This may be the proper way to look at things for stakers that believe that indeed the new minting does not really dilute their token value since it grows the platform at more than the minting rate. Stakers who are less certain of this point of view may be interested in the real or "adjusted reward rate" which subtracts the minting rate from the reward rate, getting the equality:

(adjusted reward rate)=(minting rate)/(staking rate)-(minting rate).

3.5. Staking Rate and Security

By definition, the security of a proof-of-stake platform relies on possession of tokens being the means to sybil-proofness. In other words, the consensus implied by the system is by operators that together hold a large enough fraction of tokens. Let us inspect why we would trust this type of consensus. The first reason is simply because we do not expect any malicious party to have sufficient resources to control a majority of the stake and non-malicious stakers will follow the protocol faithfully[23]. The second reason is that any set of parties that together own a large fraction of the tokens will lose a lot if the platform ceases to function faithfully since it is likely that in such circumstances the token value will drop significantly. The first argument is a typical "honest majority" argument of computer science, while the second is a game-theoretic "incentives" argument found in economics.

Quantifying each of these two reasons for security is a rather imprecise exercise since each of them depends on a host of factors. For the first reason we must try to figure out what fraction of total stake taken over by malicious parties must the platform resist. For the second reason we should estimate the economic gain that can be achieved by a malicious coalition overtaking a majority of

staking power[24] and the economic loss from drop in token's value that we expect to result from such manipulation. While exact quantification is difficult, in both cases it seems that malicious ownership of a majority of staked tokens may lead to gains that are in the order of magnitude of a constant fraction of the total value of all tokens. It follows that achieving reasonable security in face of malicious players requires staking by at least some constant fraction of total tokens. The exact constant required for different levels of security will certainly vary between platforms according to the possible manipulations, the token value and liquidity, the fraction of it that is locked and the nature of this locking. Still, for any specific platform one may look at the fraction of tokens staked as a proxy for the security obtained. Empirically, looking at Table 1, the staking rate of major platforms is in the range of about 20% to 70%[25], where the lowest rates are typically for the largest platforms.

The fraction of tokens staked in a proof-of-stake platform should be at least some platform-dependent constant, and the security increases as the fraction increases.

Given a desired level of security, and an estimate of the current rewards demanded by stakers, one may calculate the required minting rate[26]. Let us take as an example median values from Table 1: assume that stakers require 5% annual returns and that we desire a 50% staking rate. Then the required annual minting according to the equation is 2.5% (50%*5%). This type of calculation is static assuming that both the required staking rewards and the desired staking rate are fixed.

Let us see how this type of calculation can play out in a protocol. Generally speaking, the minting protocol gets to decide on the "block rewards" which determine the minting rate. Once a minting rate is set then stakers get to decide whether or not to stake their tokens, and then the minted tokens are essentially split among the stakers, thus providing their staking rewards. We may certainly assume that the higher the staking reward rate is, the more stakers decide to stake their tokens. Thus the rate of staking will adjust itself until the equality above is satisfied: if the rate of staking is lower than the equation demands then each staker receives higher rewards than "needed" and thus more stakers will flock to stake. If the staking rate is too high then the rewards are too low and stakers will leave. Equilibrium will only be possible when the rate of staking is such that the staking rewards are what is demanded "by the market".

Moreover, if the financial environment changes, while the protocol stays fixed, then the staking rate will adjust itself. For example if the outside interest rates increase or if alternative financial uses of the token within the platform become more appealing then stakers may demand higher rewards, which will be obtained by a decreased staking rate. Similarly, if the confidence in the future of the platform increases, then stakers will demand less rewards, and thus the staking rate will increase.

Of course, the platform need not choose a fixed minting rate "once and for all". Instead, as the platform may observe the current staking rate, it may use this information to determine the minting rate. Such a dynamic minting rate mechanism may allow finer control over the equilibrium of the staking rate and the minting rate as a function of the required staking rewards as they are observed from stakers' behavior. For example, Ethereum has defined a curve where the minting rate increases proportionally with the square root of the staking rate, hence making the reward rate decrease proportionally to the square root of the staking rate. An alternative is a dynamic protocol where minting rates are increased whenever staking rates are below the required level (and decreased when staking rates are higher than deemed necessary). In both cases, equilibrium will only be reached when the reward rate is equal to what stakers demand.

3.6. Bottom Line: Macro-tokenomics

So the bottom line of this section is that proof of stake platform with a utility token should cover any fixed costs of the platform by minting new tokens. The major part of the fixed cost may likely be the capital costs of staking which depend on the financial environment. As the platform's security depends on the rate of staking, the protocol should mint sufficiently many new tokens for achieving the desired security.

---

[1] The reader may consult the stakingrewards.com website for definitions of the columns in the table. Due to various differences between platforms, the precise numbers should likely be taken with a grain of salt and be viewed as indicative.

[2] These are the "real" rewards, after taking token inflation (minting) into account. For tokens with net minting the adjusted reward is lower than the reward, while for tokens with net burning it is higher.
[3] One may attempt specifying more precisely what exactly is the growth that increases trust in the platform and the limits to this effect, but for our purposes it suffices to assume that the relevant measures will grow more or less together.
[4] Unless, ofcourse, some other mechanism reduces the number of tokens.
[5] All these numbers should be taken with a grain of salt due to differences in the details of the platforms, measurement issues, the existence of locked tokens, etc.
[6] While the risky Bitcoin and a stable "stablecoin" are very different from each other in their goals and behavior, they both have the explicit purpose of serving as a type of "money" and so are classified here in the same way.
[7] As discussed, a security token represents a claim against some trusted issuer. The holder must either trust the issuer to honor that claim or seek specific enforcement of that claim. By contrast, a utility token is not a legally enforceable claim against a trusted party; it is a scarce unit of digital property that can trigger, according to the rules of the protocol, some automatic provision of a useful service by the protocol. When these systems are operating as intended, the token holder should not need to trust any issuer or court of law in order to obtain services from the protocol.
[8] We accept this reasoning in macroeconomic models all the time. If dollar holders were perfectly selfish they'd tolerate no rate of inflation, however given the mutual understanding that a stable rate of inflation on-balance encourages consumption and grows the economy, dollar holders tolerate and may even support inflationary policies because they lead to better private as well as social welfare outcomes in the long run.
[9] The "capacity" of the platform is typically defined by the protocol in a way that ensures that many operators with reasonable computing power can participate in its operation.
[10] The reader that is concerned that marginal cost pricing does not cover the fixed costs of the operation of the platform and is thus untenable is advised to wait for the section on macro-tokenomics where a source for covering these is suggested.
[11] More precisely we may want to look at the first rejected transaction as i, but our rough analysis here views these as essentially identical.*
[12] This algorithm may be viewed as rounding the linear programming relaxation of the optimization problem. Its gap from the optimal is bounded by the fraction of the total capacity of the first rejected transaction, and so it

provides a good approximation as long as transactions are relatively small compared to the total capacity. The gas price may be naturally viewed as a dual variable of this LP. Ignoring integrality constraints, maximizing $\sum_{i \in S}(v_i - c_i) = \sum_{i \in S}(v_i - \alpha s_i)$ subject to $\sum_{i \in S} s_i \leq K$ indeed happens when S contains the transactions with highest value of v_i/s_i ' with the threshold being at g.

[13] Our assumption here is that the system's costs $c_i = \alpha s_i$ for a transaction are not borne by the specific operator that decides on the block but rather by the aggregate operators and that the specific operator's costs are negligible. In cases that they are not, the "proper" mechanism should compensate the operator that decides on the block.

[14] As a first approximation, the EIP-1599 protocol dynamically finds a posted price for gas that well approximates the next block's marginal congestion costs.

[15] A good starting point for the literature is Tim Roughgarden's paper "Transaction Fee Mechanism Design".

[16] However if congestion costs are sufficiently large then these (which are part of the fees but are not actual costs borne by operators of the platform) may cover the fixed costs at least partially. This is happening to some extent in Ethereum.

[17] Perhaps with the controversial exception of public projects that may be funded by a government that takes loans that are in effect secured by the belief that the economy and future taxes will grow.

[18] If we denote by r the minting rate and denote by g the growth rate of the platform then paying fixed costs by minting is sustainable as long as $r < g$.

[19] Depending on the fee mechanism used, these congestion fees may either go directly to operators and cover their costs or be burned thus offsetting the new minting.

[20] Still, we would normally expect the operators that "sign-off" on a block to actually check the leader and verify that the block is proper and this may require some effort that we would need to incentivize. In some systems, this may be non-trivial at a microeconomic level, but it does not seem to be a significant constraint macroeconomically.

[21] Just for concreteness, here's a calculation that may serve as a very simplistic starting point for such a calculation: assume that the platform is "expected" to grow annually by g%, the minting rate is expected to be r%, the outside real interest rate is R%, and the risk premium of staking the token is estimated to be y%, then we may expect the required staking reward to be (R-g+r+y)%. More realistic financial calculations will naturally emphasize the modeling of the risk premium and the growth rate.

[22] When there are significant fixed costs beyond the capital costs then these should also be paid from new minting and the protocol will have to specify how this is done. The equation above will be modified by adding an appropriate term to make the minting rate include also these costs.
[23] This assumption may be to be considered a bit optimistic when "staking pools" concentrate the power of many independent stakers, and the staking pool may be malicious even though individual stakers are not.
[24] In some platforms a malicious majority can simply "steal all the tokens from everyone" and in other cases only more complex schemes can give a significant gain. E.g., even if a malicious majority can only effectively shut the system down then this can be used for significant gains just by shorting the token outside of the platform, where possible.
[25] These numbers should be taken with a grain of salt, e.g., since in some cases there are "locked" tokens that may only be used for staking.
[26] If the protocol also burns tokens, then this burning needs to be taken into account when looking at the net minting rate.