The crypto ecosystem has evolved into a formidable channel for raising venture capital. Each new wave of capital inflows has been epitomized by a new type of investment vehicle, may it be ICOs, DAOs, or NFTs. Regrettably, none of these paradigms tried to address the issue of investor protection, a pillar of efficient capital markets. Moreover, very few projects tried to generate economic revenue, focusing instead on marketing alone to attract new investors. Without revenues, price discovery was impossible, while investors were left without any protection against rug pulls. This has forced regulators to take a hard-line approach to the ecosystem, and rule that certain tokens are securities when they are not intended to be. Regulators have nonetheless left the door open to cryptocurrencies with truly decentralised activity like Ethereum, most notably the SEC in its interpretation of the Howey test for digital assets. We believe that a great number of decentralised projects could benefit from this regulatory exception. A system where project revenue is automatically directed to a treasury pool, and the price of tokens is computed following a predetermined bonding curve, would allow founders to efficiently raise capital, while investors would have automatic guarantees of fair participation in the success of the project. Such a framework would greatly incentivise founders to design truly decentralised projects that create value instead of hype, while making the application of securities laws less stringent or even needed. NFT royalties in particular are a pristine example of decentralised economic activity that generates cash flows, that can be used to back the value of associated tokens. We propose a cryptographic system that ties the price of tokens to the success of a decentralised activity, guarantees the fair distribution of tokens, and rewards founders and participants in the system in line with the amount of risk they are taking.

**Keywords** Smart Contract · NFT · Web 3.0

## 1 Introduction

### 1.1 A primer on traditional financial assets

Traditional financial assets have value because they confer to their owners rights and claims on underlying assets or cash flows. For instance, a corporate bond has value because it bounds the issuer to repay its debt at par at a fixed time in the future, along with periodic coupons. The price of the bond is directly linked to prevalent market interest rates and the issuer’s ability to repay. A company’s stock has value because it confers the stockholder the right to vote at the company’s general assembly, the stockholder has a claim to all future dividends, and on the company’s underlying assets. For instance, even if the company is loss-making, its assets such as patents or machinery can be sold, or the whole company might be acquired by a competitor, in which case the acquiring company would have to buy out all the shareholders. Traditional financial assets also have value because they are ruled by contracts: should the issuer renege on his legal obligations towards its shareholders and creditors, a court of law has the power to force the issuer to comply. It’s virtually impossible for a CEO to “rug pull” the company’s shareholders without facing consequences. Financial laws are harsh and far-reaching, for investor protection is paramount: it reduces financing costs, and greatly contributes to economic growth. The drawback of this traditional financial framework is, of course, the high cost of regulatory compliance.
1.2 The inconvenient truth about digital assets

Digital (or cryptographic) assets have so far struggled to achieve any of the three fundamental features of traditional financial assets:

- They confer no rights, or no valuable rights. Even in instances where a voting mechanism is built-in (DAOs, Tezos), or arranged off-chain (Ethereum, Bitcoin), holders have virtually no say on substantial issues. The voting process is hijacked by insiders (whales), or no votes on real issues are offered. Voting in crypto is very much like referendums in the USSR: people are never offered the option they actually want, and their vote isn’t counted anyway;

- They don’t grant claims on valuable assets. Because there is no link between crypto and the real world, tokens may only confer ownership claims on other tokens or coins in an exercise of pointless financial engineering;

- They don’t confer claims to future cashflows. The only notable economic revenues in the crypto ecosystem come from trading fees; most of these are generated by centralised exchanges, and are distributed to their shareholders in the “traditional finance” sense. Decentralised exchanges don’t generate cash flows as much as they reward liquidity providers for the risk of impermanent loss. NFT trading fees (a.k.a. royalties), finally, are sent directly to a cryptographic address, and any redistribution happens in centralised fashion. Absent any material sources of revenue in the so-called “decentralised economy”, there are simply no cash flows tokens could give claim to. Any financial gains to existing holders can only come from new investors, or from losses incurred by others, such as liquidation fees.

One notable exception is Ethereum. Since the EIP-1559 hardfork, instead of all the transaction fees going to miners, a part of these fees is burned. Even though ETH holders aren’t granted any explicit rights or claims, they benefit financially from overall Ethereum activity, as network users buy ETH to pay for gas. This is equivalent to stock buybacks in traditional finance, a way to return capital to share (or token) holders. However, Ethereum’s tokenomics are hard to replicate for non-infrastructure projects, both in theory and in practice. Furthermore, in some geographies like the US, securities legislation makes it almost impossible to raise financing for the development of a project by selling tokens that grant access to a service (like ETH in this case), as the service must be up and running at the time of the sale.

As certain regulators, particularly in the US and EU, are adopting a hardcore stance towards digital assets, where most tokens and coins are on the verge of being classified as securities or “quasi-securities”, the crypto ecosystem seems to be on the verge of losing its appeal as a cheap source for venture funding.

1.3 Digital assets can be the answer

Regulators have nonetheless acknowledged that the decentralisation of economic activity and governance can indeed be a new paradigm. The SEC again, in its Framework for “Investment Contract” Analysis of Digital Assets ) [SEC, 2019], points out that should the success of a project rely mostly on unaffiliated parties, in true “network effect” fashion, the associated tokens or coins wouldn’t pass the “Reliance on the Efforts of Others” bit of the Howey test. Ethereum is a pristine example of this stance: it isn’t a security because of the effective decentralisation of economic activity, independent from the acts of the founders of the cryptocurrency. The SEC has thus explicitly opened the door to a new form of fund raising, where old rules wouldn’t apply. This is particularly promising for Web 3.0 projects, that intend to create tangible value and monetise services, in a break from pure hype cycles that have characterised the crypto ecosystem up until now.

The intention of the Rogue Protocol is to build a framework for value creation that makes use of this regulatory stance. It offers a hard-coded, standalone, on-chain mechanism creating economic value reflecting the success of a decentralised economic activity. The belief is that if participants in the network can expect financial gains solely if a project is economically successful, founders will be incentivised to work on projects with sound business plans, and bring them to fruition.

2 Operational and Technical Considerations

We set out to build a standard for Web 3.0 value creation rewarding token holders from decentralized economic activity. We want this mechanism to be as direct as possible, minimizing fees, and avoiding the direct use of external DeFi architecture or any centralised services.
2.1 Returning capital to token holders (old method)

There are two ways to funnel the profits of economic activity to investors: by paying them dividends, or by buying back their “proof of investment” certificates, may they be shares, IOUs, or cryptographic tokens. Paying dividends with smart contracts is both impractical, because of on-chain fees, and complicated, due to the limitations of smart contract programming languages (a smart contract can’t natively access the list of token holders, for example). Buying back tokens on exchanges is prone to front-running and slippage if liquidity is low, and requires trusting third parties, which can fail. Moreover, requiring tokens to be listed on exchanges imposes additional costs on founders, who must pay kickbacks to centralised exchanges, and commit material liquidity for market making, which diverts substantial financial resources away from operations.

2.2 Market making with bonding curves and treasury pools (new method)

Instead of relying on exchanges (decentralised or not) as a primary solution for enabling the buying and selling of tokens, we propose the use of a smart contract that acts as a market maker. The Rogue smart contract manages treasury pools for every token, where it owns both a reserve of tokens, and a reserve of liquidity (ETH on Ethereum, BSC on Binance Chain). The price at which the smart contract buys and sells tokens is a function of the amount of liquidity in the treasury pool. This function is the bonding curve of the token. Tokens can of course be listed and traded on exchanges, but the smart contract acts as a fallback solution, or a “buyer and seller of last resort”. The bonding curve must be bound by the following conditions:

a) The price of tokens when there is no liquidity in the treasury pool, is zero;
b) The first derivative of the bonding curve must be positive (price increases along with liquidity in the treasury pool);
c) The second derivative of the bonding curve must trend towards zero (the bonding curve can’t be parabolic towards infinity).

Condition c) is often overlooked. We have seen many instances of parabolic, or even exponential bonding curves, designed to attract investors who are looking for digital assets that “go to the moon”. However, such bonding curves are unstable, and inevitably lead to “rug pulls”, as initial investors can obliterate the price by selling a relatively small stake of tokens. Instead, a linear regime allows a “whale” to sell a large stake (for example, 20% of all tokens) with a manageable decrease in price (for example, 5%).

2.3 Ownership of tokens

Traditionally, project founders pre-mint a large amount of tokens, which they own in full, and sell these tokens to investors, at varying prices depending on the maturity of the project. This arrangement is a carbon copy of traditional finance and corporate structure, where founders are initially whole owners of their company, and get progressively diluted as they raise capital. We don’t see any fundamental reasons for this model to be valid in fundraising for decentralised projects. The main reason is that unlike equity in a company, cryptographic tokens don’t confer rights or claims; they are solely a “proof of token ownership”. Project founders are fundamentally just as free to act whether they own 90% of tokens, or none at all. The only factor guiding their efforts is financial incentives, namely:

a) Structuring their project in such a way that attracts investors;
b) Granting the founders additional financial gains in case of success, i.e. aligning investor and founder incentives post-funding.

Condition b) implies that founders should own a substantial amount of tokens. How should they gain control of these tokens? Mathematically, a bonding curve binds the price of tokens to the amount of liquidity in the treasury pool. This relationship can also be seen as binding the price of tokens to the amount of tokens in circulation, i.e. the amount of tokens issued. Thus, we move that at mint, as the associated treasury pool is empty, all the tokens are owned by the smart contract, and project founders must buy some tokens at a low price, thus seeding the treasury pool, and enabling the smart contract to act as market maker.

2.4 Token minting

Anyone can mint a treasury pool, by paying the associated gas fee, and choosing the bonding curve parameters (elasticity and amplitude). Tokens are minted in a state of zero-ownership, and the issuer (from a legal and practical point of view) is the smart contract itself. Founders must then buy an initial stake of tokens, along with the pool’s naming rights.
Such a mechanism for issuing tokens would fall outside the remit of existing securities regulation in most jurisdictions. [Hinman, 2018] [Sygna, 2021]

### 2.5 Programmatically rewarding participants

The market-making smart contract has two core functions for depositing liquidity into the treasury pool (buying tokens), and withdrawing liquidity (selling tokens). Depositing liquidity increases the price of tokens according to the bonding curve, and vice-versa. A decentralised project that wishes to abide by the framework, can direct a portion of the fees it generates to its treasury pool, thus rewarding participants. Note that this mechanism is intended to be completely automated, and on-chain. The function for depositing liquidity has a parameter, where the caller signals to the smart contract if it wants to get tokens in return, or not. A decentralised project running smart contracts can thus deposit liquidity without any new tokens being issued, and explicitly give away the liquidity it has deposited into the treasury pool, to token holders. This is default setting: if someone simply sends liquidity to a pool’s treasury address, no new tokens are issued. As an illustration of this principle, the smart contract itself takes a small fee on each transaction, and deposits it into a Rogue Treasury Pool. We have intentionally given up the control on this smart contract, programmatically binding the revenues flowing into this pool into perpetuity. This mechanism makes a strong case that the Rogue Protocol itself qualifies as truly decentralized.

### 2.6 Tokenising NFT royalties

Currently, NFT royalties (or NFT trading fees) are hard-coded into NFT smart contracts, including a (list of) cryptographic address(es) where those royalties should be sent to each time the owner of the NFT changes. These royalties are intended to compensate the issuer of the NFT for his work, on top of the initial mint price. NFT traders oftentimes see this as a transaction tax, and NFT creators are under huge pressure to set the royalty rate as low as possible. At the same time, this arrangement as a mechanism for rewarding creators and artists isn’t efficient:

1. Creators and artists might have to wait for a very long time to recoup their costs;
2. Setting a cryptographic address as the beneficiary bears a huge operational risk, as the creator or the artist might lose their keys, and the royalties will be sent into oblivion, into perpetuity.

We offer a different solution, where royalties are sent to a treasury pool, and anyone can buy or sell tokens. This enables NFT collectors and even outsiders to participate in the success of an NFT collection, while greatly mitigating operational risks. This mechanism is well known in finance as “securitization”: cash flows from various sources (usually mortgages, bonds or loans) are contractually redirected to a pool, and investors can acquire bonds backed by these cash flows. The tokens of such a royalty pool have fundamental value: the pool is “worth” the present value of all expected royalties payments. They can be traded, transferred, or redeemed. Tokenising an NFT’s royalties creates a completely new digital asset class: royalty tokens.

### 3 API Specifications

On top of the standard BEP-20 and/or ERC-20 token functions, these are the public functions implemented by the Rogue Protocol.

#### 3.1 TreasuryPool

##### 3.1.1 deposit(address account, bool issueTokens)

Deposits liquidity to a pool (BSC for BEP-20, ETH for ERC-20), and issues pool claim tokens to the sender while transferring a 0.2% commission to the Rogue Treasury Pool. The amount of claim tokens issued depends on the amount of claim tokens previously issued, and of the pool’s elasticity K and amplitude M parameters, according to a bonding curve formula following three different regimes:

\[
M \left( \frac{x}{A} \right)^K \{ 0 \leq x \leq A \} \\
M (2 - (2 - \frac{x}{A})^K) \{ A < x \leq B \} \\
\frac{M}{B} (2 - (2 - \frac{B}{A})^K) x \{ x > B \}
\]
where \( B \) is the solution to the equation:

\[
(2 - \frac{x}{A})^{(K-1)}(2 - \frac{x}{A} + \frac{Kx}{A}) = 2
\]

simply put, \( B \) is the amount of claim tokens issued where the slope of the bonding curve is tangent to a linear function (which becomes the bonding curve for the third regime).

Under the first regime, the amount of liquidity needed to issue \( x \) claim tokens is equal to:

\[
L_1(x) = \frac{MA}{K+1} \left( \frac{x}{A} \right)^{(K+1)}
\]

very simply, this is the integral of the bonding curve function under the first regime. The maximum of liquidity than can be deposited under the first regime is thus:

\[
Max_{(regime_1)} = \frac{MA}{K+1}
\]

Under the second regime, the amount of liquidity needed to issue \( x \) claim tokens is equal to:

\[
L_2(x) = 2Mx + \frac{AM}{K+1} \left( 2 - \frac{x}{A} \right)^{(K+1)} - MA \left( 2 + \frac{1}{(K+1)} \right)
\]

and the maximum of liquidity than can be deposited under the second regime is:

\[
Max_{(regime_2)} = \frac{AM}{K+1} \left( (2 - \frac{B}{A})^{(K+1)} - 1 \right) + 2M(B - A)
\]

Under the third regime, the amount of liquidity needed to issue \( x \) claim tokens is equal to:

\[
L_3(x) = \frac{M}{2B} \left( 2 - \left( 2 - \frac{B}{A} \right)^{K} \right) \left( x^2 - B^2 \right)
\]

and there are no maximum constraints on liquidity or claim tokens in the third regime.

N.B. : The amounts of claim tokens to be issued are calculated by solving for \( x \) (liquidity to be deposited) the equations above, taking into account the amount of claim tokens already deposited. Approximations have to be made, as these polynomial equations don’t have a general solution for \( K > 2 \). In practice, the computation is done using the bisection method, using iterative functions. A compromise between the accuracy of computations, and gas costs, has to be made.

The calculation takes into account the changes of bonding curve regimes, in case a deposit produces an overlap of regimes, to prevent obvious arbitrage/hacking possibilities.

If the issueTokens parameter is set to false, no claim tokens are issued to the sender. If the pool’s liquidity exceeds the theoretical liquidity that should be in the pool according to the amount of claim tokens already deposited. Approxi...
3.1.2 withdraw(address payable account, uint256 amount)

Withdraws liquidity from the pool, according to the amount number of claim tokens submitted for withdrawal. The amounts of liquidity to be withdrawn can be calculated exactly. The calculation takes into account the changes of bonding curve regimes, in case a withdrawal produces an overlap of regimes, to prevent obvious arbitrage/hacking possibilities.

N.B.: the liquidity deposited into a pool’s treasury can diverge from the theoretical liquidity that should be in the pool’s treasury according to the amount of claim tokens issued. This is due to:

1. rounding errors while calculating deposit/withdrawals (minor);
2. deposits that don’t require claim tokens to be issued, which is what happens when royalties are sent to the pool’s treasury following an NFT trade.

As a consequence, the amounts of claims tokens issued during deposits, or liquidity withdrawn in exchange of claim tokens, are adjusted by the following factor:

$$\beta(\text{claimtokensissued}) = \frac{\text{actualliquidityinpool}}{\text{theoreticalliquidityinpool}}$$

(10)

This is to ensure that the system is stable, and that no liquidity is locked in forever by the Rogue Protocol in a pool.

3.1.3 function generatePool(int256 _m, int256 _k, string memory _name, string memory _symbol, string memory _url, string memory _iconUrl) public payable

Mints a new pool with the parameters m (amplitude) and k (elasticity), and deposits an initial amount of liquidity into the pool’s treasury, returning the sender the ad hoc amount of pool tokens. The sender is also granted the naming rights to the pool for a short period of time, so that he can set the pool’s description parameters.

3.2 TPNamingService

Treasury pools are created in a nameless state, with a generic name and icon. Anyone can buy the right to name and link to an icon any nameless pool, by paying a naming fee. Pool names are unique - if a pool with an identical name or ticker (non-case sensitive) already exists, the naming transaction fails.

3.2.1 namePool(address pool, string memory name, string memory symbol, string memory iconUrl)

If a pool’s naming rights aren’t taken, this function allows the sender to buy the naming rights to the pool, set a name, ticker, and icon url, for a period T dependent on the current naming rate $\alpha$:

$$\alpha T = N_F$$

(11)

where $N_F$ is the naming fee.

$\alpha$ is variable to account for the popularity of the naming system: if the rate is too high relatively to the perceived value of naming pools, and less pools are being named, $\alpha$ goes down. Note that once calculated for a given naming transaction, $T$ remains the same and doesn’t decrease/increase as $\alpha$ increases/decreases. All the naming fees are deposited by the Rogue Protocol into the Rogue Treasury Pool. The owner of the naming rights can add more naming fees before T expires, at the prevalent $\alpha$, or let his/her naming rights expire. The owner of the naming rights can also change the name and/or icon of the pool at any time, under the condition that the new name isn’t taken. $\alpha$ is recalculated every time a new pool is minted, following the formula:

$$\alpha_n = \frac{(NP_n + 1)^2 (AP_{n-1} + 1)^2}{(NP_{n-1} + 1)^2 (AP_n + 1)^2} \alpha_{n-1}$$

(12)

where $NP_n$ is the number of named pools at the start of the nth period, and $AP_n$ is the number of anonymous pools at the start of the nth period. Thus, if the proportion of named pools increases, the current naming rate increases, and vice versa. If no new pools are named, as the number of pools increases through the minting process, or naming rights expire, the current naming rate decreases until a new economic equilibrium is found. We use a convex formula to counter the natural tendency of this series to trend towards zero. Note that the possibility to add a variable naming fee allows for naming rights owners to lock in low naming rates for their pools, or only pay a minimal fee in anticipation of a rate drop.
3.2.2 `swapNamingRights(address pool1, address pool2, bool name, bool symbol, bool fees)`

Allows someone who owns the naming rights to both pools, to swap those rights. This allows to buy a pool with a specific name/ticker, and to transfer that name/ticker to another pool.

3.2.3 `setAutoTransferPrice(address pool, uint256 price)`

Allows the owner of a pool’s naming rights to set a price for which he/she is willing to sell those naming rights.

3.2.4 `renamePool(address pool, string memory name, string memory symbol, string memory url)`

Allows the owner of a pool’s naming rights to change its name, ticker, or icon url.

3.2.5 `buyNamingRightsByPool(address pool)`

Allows anyone to buy the naming rights to a pool whose auto transfer price has been set by its owner, given the pool’s address. A commission fee of 0.2% will be sent to the Rogue Treasury Pool.

3.2.6 `buyNamingRightsByName(string memory name)`

Allows anyone to buy the naming rights to a pool whose auto transfer price has been set by its owner, given the pool’s name.

3.2.7 `buyNamingRightsBySymbol(string memory symbol)`

Allows anyone to buy the naming rights to a pool whose auto transfer price has been set by its owner, given the pool’s ticker.

3.2.8 `extendExpiryDate(address pool)`

If the sender is the owner of the naming rights to the pool, extends the naming period according to the amount of fees sent in the transaction.

4 Decentralised Market Making & Alignment of Incentives

4.1 The issue with automated market making and liquidity pools

Decentralised exchanges use liquidity pools as an alternative to order books for market making, which achieves two objectives:

a) it allows for market making in an environment of high transaction costs (i.e. gas fees);

b) it allows for holders of cryptocurrencies to earn interest on these cryptocurrencies by locking them in liquidity pools.

In a sense, liquidity pools for automated market making are a mechanism for transferring trading commissions from traders to liquidity providers. However, such an arrangement only works well for liquidity providers only so long as the prices of assets in the liquidity pools don’t move too much relatively to one another, and trading fees compensate liquidity providers for the risk of impermanent loss. Studies show that on aggregate, liquidity providers are better off not contributing liquidity, as the risk of impermanent loss is empirically too high relatively to their cut of trading fees [Loesch et al., 2021]. Liquidity providers thus aren’t incentivised to add uncorrelated and/or undervalued assets to liquidity pools, thus making those assets illiquid. In microeconomics jargon, this means that undervalued, uncorrelated assets end up with a high price elasticity factor, which impedes on efficient price discovery. As a result, the liquidity pool model creates a pit of illiquidity for upstart projects, which can only be solved by insiders committing and withdrawing large amounts of their own holdings. This amounts to insider trading and market manipulation, and introduces the risk of regulatory clampdown, putting the project and its founders at risk. Moreover, as insiders de facto control liquidity pools and set the price of their cryptocurrency, third parties are strongly disincentivised to contribute to the market-making process, raising the funding costs of any project.
4.2 Predetermined bonding curves for low-cost on-chain trading

We propose a different approach, using a predetermined bonding curve model that mimics the business creation cycle:

At mint, a smart contract owns all the claim tokens of an empty treasury pool. Founders and investors can buy tokens by depositing liquidity into the treasury pool, or withdraw liquidity by selling their claim tokens. The smart contract calculates the price of the tokens for the purposes of each transaction according to the amount of claim tokens previously issued (or, equivalently, according to the amount of liquidity already present in the treasury pool). We have designed a generic formula for Rogue Protocol’s bonding curves (see 5. Generating Bonding Curves) that takes as input two parameters, and produces as output a bonding curve that aligns the incentives of distinct economic actors according to three different regimes of price behavior:

a) founders and early backers “get in cheap”, as they bear most of the risk of the project’s failure;

b) speculators get in when the price starts to rise fueled by early profits, and price elasticity gradually goes down as the project goes through the “price discovery” phase;

c) investors get in as price is high, but volatility is low, as the project’s profit stream stabilises, and the fundamental risk of failure becomes low.

4.3 Economic rationale: the J-Curve

“The cash flow of a Private Equity investment has a negative return or cash flow during the initial years, usually during the first 3 to 5 years. After this period the project should produce positive returns and the investment is expected to begin to have increased evaluations of the underlying investments. The chart shows how cash flows in Private Equity, when plotted against time, show a J shaped curve.” [D’Alessio 2015]

The J-Curve effect is well known to Venture Capital and entrepreneurs alike. Startups experience an initial phase of high investment and low revenue, while their chances of ultimate success are very uncertain. To compensate for the risk, founders and early investors "get in cheap" - investors buy shares at low prices as compared to the expected valuation of the business should it succeed, while founders get into the capitalization table for free - but they have make up for it during the early years, by working nights and weekends while being paid only a fraction of what they could make at an established company. The bonding curve governing the price of claim tokens is intended to reflect this empirical law.

Each business is different, and its risk/reward profile is unique. Thus, to accommodate for the needs of the quickly evolving Web 3.0 economy, we propose that treasury pools should be minted with bonding curves based on two factors:

1. The elasticity factor K, which defines how the claim tokens’ price elasticity varies;
2. The magnitude factor \( M \), which defines the amplitude of price increase in case of success.

![Diagram showing bonding curves](image)

Founders and early investors can deposit liquidity into the pool’s treasury to receive a fraction of its claim tokens at a “founders’ price”, and then work towards the success of their project, in order to cash out once the project has grown to a “cash generation phase” and investors move in, providing ample liquidity. This structure of the bonding curve allows founders to cash out without crushing the price of the claim tokens, thus reducing the risk for late investors, and increasing overall economic value. Founders are incentivised to pick adequate elasticity and magnitude factors:

1. if the magnitude factor is too low relative to the project’s risk/reward profile, the founders won’t be adequately rewarded for their work;
2. if the magnitude factor is too high relative to the project’s risk/reward profile, the founders will fail to reach the “cash generation phase”, and will not be able to cash out as they won’t manage to attract late investors;
3. if the elasticity factor is too low relatively to the probability of success, speculators and late investors will get a free ride on the backs of the founders;
4. if the elasticity factor is too high relatively to the probability of success, speculators and late investors won’t buy in, and the founders won’t be able to cash out.

By minting a pool with the appropriate magnitude and elasticity parameters, founders can effectively and efficiently signal their true intentions to market participants, empowering them to act in an informed manner. The fact that those parameters can’t be changed ties the early participants to a set of ironclad rules that protect the interests of non-insiders. Each token is minted with a state of zero-ownership, meaning that its 1,000,000,000 claim tokens are owned by the Rogue smart contract itself. Should all the initial 1,000,000,000 claim tokens have been depleted, the smart contract will automatically issue additional claim tokens.

5 Generating Bonding Curves

To generate the bonding curve described above with a smart contract, one must take into account the limitations of smart contract programming languages. For instance, Teal and Solidity don’t have trigonometric functions, and Ethereum doesn’t support floating point numbers. One must also take into account the gas price for calculating the price of claim tokens each time a deposit/withdrawal is performed. As a result, we have opted for the bonding curve formulas described in [3].

---

[3] Reference to the source of the bonding curve formulas.
Each bonding curve has three regimes: a "parabolic growth" regime from 0 to A (A = 400,000,000 is arbitrarily defined as the critical mass of issued claim tokens beyond which the success of the project becomes almost certain), a "parabolic sloping down" regime from A to B, and a "linear growth" regime beyond B. B is the spot where the inverted parabolic curve (2nd regime) is tangent to a linear function (3rd regime), and is calculated once, by dichotomy with A < B < 2A, when the pool is minted.

With M=100,000,000 (an unrealistically high magnitude factor for the sake of making the shape of the curve apparent), we obtain the following curves:

Another way to look at these bonding curves, is to consider the expected returns of founders and early investors, should the project succeed, i.e. should the amount of claim tokens issued grow beyond A – may it be by generating profits and redirecting it to the pool, or by attracting enough deposits by any other means. Let’s assume founders deposit the amount of liquidity needed to receive the first 200M tokens, early investors receive the next 100M claim tokens, and they all cash out at the “point of proven success” A. Let’s set M = 0.0001. We’re assuming founders cash out 20,000 liquidity coins, and early investors cash out 10,000 liquidity coins.

K = 2: Founder deposited liquidity: 1666, founder return: 1,100% Early investor deposited liquidity: 3958, early investor return: 152%
K = 5: Founder deposited liquidity: 104, founder return: 19,100% Early investor deposited liquidity: 1082, early investor return: 823%
K = 10: Founder deposited liquidity: 1.8, founder return: 1,126,300% Early investor deposited liquidity: 152, early investor return: 6,487%
K = 20: Founder deposited liquidity: 0.001, founder return: 2,202,009,500% Early investor deposited liquidity: 4.5, early investor return: 220,680%
K = 30: Founder deposited liquidity: 0.000006, founder return: 3,328,600,000,000% Early investor deposited liquidity: 0.17, early investor return: 5,786,241%

Early investor returns on Amazon, for instance, point to an elasticity factor of K = 27. Miguel Bezos, Jeff’s father, bought 582,528 shares of Amazon at a price of $0.1717 in February 1995 [SEC 1997]. This puts his current return on investment at a little over 2,000,000%. The amplitude factor M is difficult to determine, for it depends on the price of the underlying liquidity coins (which have a much higher volatility than fiat). But with all else being equal, BSC
trading at $500, and Amazon’s market cap standing at $2T, we obtain $M = 10$ on the Binance Chain. This is an extreme example, of course, as Amazon is one of the most successful investor stories in the history of finance. But it provides a down-to-earth illustration of how the Rogue Protocol would work in the centralised economy. In practice, the Rogue Protocol mints new pools with $K$ and $M$, with $K \in [1, 99]$ and $M \in [0.000001, 0.01]$. With ETH trading at $2000$, this allows for projects that turn out successful to be valued at between $800K$ and $8B$, a wide range that should suit most of Web 3.0 ventures. Of course, there is no upper bound to valuation – we’re simply talking orders of magnitude here (market cap at 400,000,000 claim tokens issued).

## 6 The Rogue Protocol and the Rogue Treasury Pool

The Rogue Protocol uses an autonomous, uncontrolled smart contract whose purpose is to:

1. mint new tokens with associated bonding curves;
2. process deposits into, and withdrawals from, treasury pools;
3. manage the token naming system.

To illustrate the effectiveness of this system, the Rogue Protocol has its own treasury pool, the Rogue Treasury Pool. All the revenue generated by the Rogue Protocol as an actor in the Web 3.0 economy, is programmatically deposited into the Rogue Treasury Pool, without asking to be issued tokens in return. This theoretical stream of revenue should thus support the value of the Rogue Treasury Pool’s claim tokens. Should the Rogue Protocol be a commercial success, the price of the Rogue Treasury Pool’s claim tokens would go up regardless of whether founders, early investors, speculators or late investors deposit liquidity into the Rogue Treasury Pool. The Rogue Treasury Pool is created with the following $K$ and $M$ parameters:

$$M_{RSPTreasury} = 0.01$$
$$K_{RSPTreasury} = 13$$

The tokens minted by the Rogue Protocol are enhanced standard ERC-20 and BEP-20 tokens, which makes them compatible throughout the existing crypto ecosystem. They can be transferred to wallets, traded on exchanges, staked, and more generally used in all standard DeFi projects:

The Rogue Treasury tokens minted by the Rogue Protocol (also known as Rogue Coins) do not represent any contractual or economic rights in any business enterprise but simply serve as a mechanism to allow the Rogue Protocol to function. In this sense, Rogue Treasury tokens (or Rogue Coins) are utility tokens and do not constitute securities or security tokens. Rogue Coin holders are not entitled to any ownership in, or control of, the Rogue Protocol or any related issuer. Finally, Rogue Coin does not purport to offer any yield or return or otherwise operate as an investment.
7 Conclusion

The crypto ecosystem has proven itself to be a formidable tool for capital raising. There is little doubt that it greatly contributed to shifts in societal mindset and financial regulatory policy and legislation.

However, without a proper framework to align the incentives of founders and investors, this funding tool is destined for self-destruction, as resources end up wasted, and investors fail to achieve long-term, sustainable returns. We see the Rogue Protocol as an “interesting experiment” in this direction. We certainly don’t claim to solve all of crypto’s shortcomings. But stirring up debate would already be an amazing achievement as of itself. As pointed out in [2,3], cryptographic tokens are a different breed of assets. Indeed the Law Commission of England and Wales is consulting on recognizing crypto-assets as an entirely new form of property, termed “data objects”. [Commission, 2022] They are merely “proofs of investment”, and should be treated as such. We only deal with the alignment of financial incentives between founders and investors; in a stark departure from traditional finance, investors have no say in corporate governance, and oftentimes, no recourse.

As such, cryptographic tokens as a funding tool is a paradigm that only suits specific enterprises, decentralised in nature, as outlined by financial regulators themselves. Yet the very fact that these regulators have validated the “crypto exception”, albeit in a limited way, can be seen as a crucial milestone for the future of the decentralised economy.

What we need now is proof that an alignment of incentives between founders and investors, is enough to create long-term value. We are painfully aware that without negative externalities to simply running away with investor money, bad actors will always try to game the system. But we also believe that giving founders the freedom to act as they deem best fit, without the incessant meddling of venture capitalists obsessed with short-term milestones, can be hugely beneficial.

Be humble, be honest, and try your best. Godspeed.

References

U.S. SEC. Framework for “investment contract” analysis of digital assets, 2019. URL [https://www.sec.gov/corpfin/framework-investment-contract-analysis-digital-assets].

William Hinman. Digital asset transactions: When howey met gary (plastic), 2018. URL [https://www.sec.gov/news/speech/speech-hinman-061418].

Sygna. Sec investigates uniswap: What’s next for defi regulation?, 2021. URL [https://www.sygna.io/blog/sec-investigates-uniswap-whats-next-for-defi-regulation/].

Stefan Loesch, Nate Hindman, Mark B Richardson, and Nicholas Welch. Impermanent loss in uniswap v3, 2021. URL [https://arxiv.org/abs/2111.09192].

Gino D’Alessio. Private equity and the j-curve, 2015. URL [https://sophisticatedinvestor.com/private-equity-and-j-curve/].

U.S. SEC. Amazon registration statement, 1997. URL [https://www.sec.gov/Archives/edgar/data/1018724/0000891618-97-001309.txt].

Law Commission. Digital assets consultation paper, 2022. URL [https://s3-eu-west-2.amazonaws.com/lawcom-prod-storage-11jsxou24uy7q/uploads/2022/07/Digital-Assets-Consultation-Paper-Law-Commission-1.pdf].