Spot Protocol: Trustful Non-Fungible Token Issuance

(DRAFT)

Jin Huang
jin@spotprotocol.io

Saureen Shah
saureen@spotprotocol.io

Rikin Gandhi
rikin@spotprotocol.io

https://spotprotocol.io

September 5, 2018

Abstract

The value of an asset depends on whether a buyer can trust the validity of its financial interest. This is especially challenging in decentralized exchanges in which no central authority provides such a guarantee. Spot is an open protocol to represent assets as distinguishable tokens and to enable their trustless trade. The protocol creates standards to ensure incentive compatibility among market participants, including the seller, underwriter, and buyer. Spot incorporates roles for other actors, including servicers and attestors, that are often needed for complex asset types. The protocol coordinates the tasks of each of these participants to originate, tokenize, and service digital and physical assets on the blockchain. With Spot, new and existing businesses can unlock greater value for their assets through decentralized markets.

Contents

1 Introduction **2**

1.1 Overview . . . 2

1.2 Related Work . . . 3

2 Spot Assets and Market Mechanisms **4**

3 Reputation and Incentives **5**

3.1 Technical Preliminaries . . . 6

3.2 Reputation . . . 7

3.2.1 Underwriter Reputation . . . 7

3.2.2 Servicer Reputation . . . 10

3.2.3 Buyer Reputation . . . 11

3.2.4 Asset Reputation Liability . . . 12

3.3 Incentives . . . 13

3.3.1 Asset Administrators: Underwriter and Servicer . . . 13

3.3.2 Seller . . . 14

3.3.3 Buyer . . . 15

4 Summary **15**

5 Appendix **17**

5.1 ERC-721: Tokens Permission and Exchange . . . 17

5.2 Verification Game and Reputation Score . . . 17

5.3 Decentralized Data Privacy and Security . . . 18

1. Introduction

1.1 Overview

Asset markets rely on legal and financial guarantees to facilitate exchange with buyers. Underwriters bear the risk of offering sellers' assets, and attestors, such as legal and accounting firms, ensure the validity of their origin and value. In a decentralized market, a lack of trust

among market participants compounds buyer risk, requiring additional assurances for assets that need ongoing appraisals and servicing.

Blockchain technologies could enable the trade of existing digital and physical assets, from real estate to virtual goods, that otherwise would not be easily transportable across markets. The representation of an asset as a token makes transferability low-friction through smart contracts. The transactions produce an immutable audit trail on the blockchain. A single token can be further tokenized to create parts of the original, each with its own distinct value, enabling further fractionalization of ownership. With open decentralized standards, asset management and trade can now transition from siloed institutions to more transparent exchanges, creating opportunities for sellers to expand their capital pools, buyers to gain wider investment exposure and liquidity, and new decentralized marketplaces and business models to develop.

The main challenge of asset tokenization is trust. A buyer of a tokenized asset needs to validate the legitimacy of the digital tokens against a decentralized registry. Spot provides a set of rules that specify the roles of market participants to sell, buy, underwrite, and attest assets. Each actor plays a particular role in a market which the protocol defines as tasks. Underwriters, for instance, may provide offline information pertaining to an asset's ownership records and servicers may be responsible for distributing its payments. Buyers request attestations to confirm the reliability of both, or raise a dispute. Reputation scores are maintained and updated for each participant. The protocol ensures that each actor is rewarded or penalized so that individually rational actions are collectively optimal for the marketplace.

1.2 Related Work

There are a variety of initiatives in the works to bring real-world assets to the blockchain. Mattereum is working on making smart contracts legally enforceable (Gupta, 2018). R-token defines a standard for a security token to make trade and ownership compliant with security laws (Remeika, Amano, and Sacks, 2018). Stellar is focused on larger financial institutions to help issue and trade assets on the blockchain (Mazieres, 2018). And TrustToken is attempting to tackle tokenization of all real world assets, starting with currencies. The platform uses a simple staking system as a form insurance at the amount of 5% to guarantee the tokenized assets.

Spot Protocol's risk and reputation system follows the core principles of mechanism

design. Jackson (2003) gives an introduction to the literature. The calculation of reputation uses historic chain data that is not accessible directly through the EVM. We take inspiration from Teutsch and Reitwießner (2017) and Peterson, Krug, Zoltu, Williams, and Alexander (2018) to ensure truthful revelation.

2. Spot Assets and Market Mechanisms

The foundational objective of the protocol is to ensure that a buyer can trust that tokenized assets on the blockchain are legitimate and, where necessary, properly serviced. The protocol's incentive structures and mechanisms are designed to provide this assurance specifically for decentralized marketplaces where no central authority makes a guarantee.

An asset is originated on the blockchain and represented by a non-fungible token (NFT). Since the buyer of an NFT asset in the decentralized platform does not directly communicate with the owner of the physical assets, asset buyers rely on asset administrators, underwriters and servicers, to perform tasks truthfully to bring information on-chain. The protocol specifies how asset administrators are incentivized to perform these tasks, how attestation and dispute mechanisms can be requested to corroborate the validity of asset information, and how staking and reputation serve as a check and balance of all market actors. In this section, we focus on describing the life cycle of assets, and the actions that the protocol enables.

Tasks represent the fundamental unit of work performed on assets. A task specifies the action that is executed on an asset, the market actor responsible for it, and their reward for doing so. Asset classes may differ in the number and type of tasks performed on them.

The lifecycle of an asset on a decentralized marketplace begins when an underwriter creates the first task event to originate an asset as an NFT. The seller and underwriter structure the terms of a contract to specify an NFT holder's financial interest into the actual asset. This contract is registered as an NFT on the protocol's asset registry. A buyer invests and becomes the new owner of this token. The token can be further bought and sold. An NFT is removed from circulation when a termination task event is marked closed. In between the origination and the termination event, there are other task events that are created, performed, and closed. The complete collection of task events create an immutable history for an asset.

Task events involve multiple stages, but have a minimum of two: created and closed. An asset's first event is an underwriting task. The event's status is marked as created when

the NFT is originated and when a buyer pays for the asset, the event is marked as closed.

The servicing task event involves three stages: created, performed, and closed. An NFT representing a loan, for instance, requires a servicer to distribute monthly payments to the NFT token holder. Servicing tasks are created when the NFT is issued. The tasks are performed by the servicer at the end of every month. The task is closed when the buyer verifies that the payment is correct.

Buyers can dispute the performance of a task event before it is marked closed. A servicer, for instance, may be tasked with collecting repayment of the principal of a loan asset. A buyer could request a dispute task to contest or attestation task to validate whether the servicer truthfully distributed the repayment. To initiate a dispute, the buyer stakes tokens. The dispute is then resolved in one of three ways:

- First, the dispute could be verified as a legitimate issue by the seller. Both the seller and buyer would then receive a reward, and the servicer's reputation and staked tokens would be penalized.
- Second, the dispute task could expire. In that case, the buyer's staked tokens would be burned, and servicer's performance reward would be diminished. The expired dispute would reduce the reputations of both the buyer and servicer.
- Third, the dispute could be withdrawn. The buyer and servicer could communicate outside of the marketplace, off of the blockchain, to resolve their disagreement. See section 3.3 for detailed discussions on these incentive structures.

The protocol includes the ability to request attestations to provide an additional level of credibility for the performance of task events. An underwriter, for example, may publish a financial statement of a seller's business operations, but may task a third party accounting firm to conduct an audit. Similarly, buyers may request attestations to confirm the credibility of information pertaining to an asset originated by an underwriter who has a low reputation. Attestation tasks include a reward structure in which an attestor, like an accounting firm, takes the request and submits an attestation report. The attestor is rewarded when the buyer ultimately accepts the report.

3. Reputation and Incentives

In this section we describe the details of the protocol's incentive and reputation system for the coordination of participant actions to optimize market outcomes.

3.1 Technical Preliminaries

Let A represent a set of NFT assets. For any asset $a \in A$, we use ι_a^1 to represent the seller of the asset. The superscript set $\{1, 2, 3, 4\}$ is used to represent the underwriter, servicer, seller, and buyer. The underwriter is denoted by ι_a^1. The identities of the underwriter and seller are immutable. If asset a's ownership is transferred in a secondary market, ι_a^4 represents the current owner of the asset. The role of a servicer of an asset ι_a^2 is transferable.

Each asset belongs to an asset class c. Let $h(\iota, c, t)$ be the amount of Spot Protocol Token[1] (SPT) agent ι can stake, which are locked for a particular time t. Each asset class has a particular risk rate β_a. This could be considered as a default rate or probability of an exceptional event of the underlying asset. The risk rate is used to normalize liability across asset classes.

A task event is denoted by e. Marketplace agents create tasks for an asset. E_a denotes the set of task events associated with the asset a. $ET_a \subset E_a$ represents the subset of tasks related to an attestation request. Every asset has at least one task event associated with its creation and termination. Let $\tilde{e}_a$ be the creation event associated with asset a and $\bar{e}_a$ be the termination event.

Let $\xi_\gamma(x)$ be a scaling curve,

$$\xi_\gamma(x) = \frac{1}{\int_0^\infty e^{-\gamma t^2} dt} \int_0^x e^{-\gamma t^2} dt. \tag{3.1}$$

This curve is used to normalize a cumulative effect, which is defined in the description of the reputation formula. γ is a decay rate. Note that $\xi \in [0, 1]$, see figure 1. For example, as the transactions of a seller increase, the cumulative effect scales by ξ with a bounded maximum.

Let ψ be the adjusted price of an asset,

$$\psi(a) = p_a \left(1 + \int_0^{T_a} \frac{1}{(1+\tau_a)^t} dt\right), \tag{3.2}$$

where p_a is the initial price of asset a, T_a is the maturity duration of the asset, and τ_a is the inverse discount rate. For an asset class c, τ_a is equivalent to any $a \in A(c)$. That is, the inverse discount rate is fixed for each asset class. ψ is used as the relative weight of each asset. For example, the transaction of two assets sold for the same price could have varying bearing on reputation based on their maturity duration. An asset class with a large τ is similar to

[1] Spot Protocol token is also known as SPT. It could also be referred to as the protocol token.

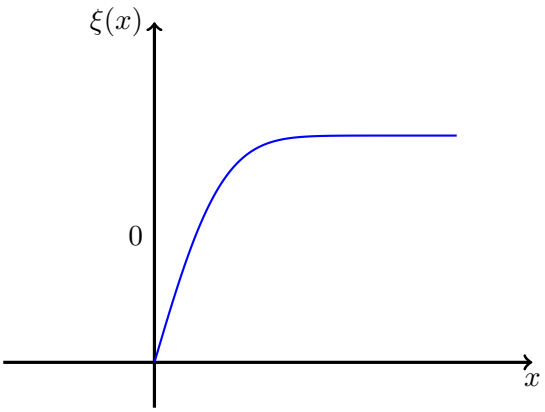

Figure 1: An example of the function $\xi(x)$.

the adjusted price being the same as the initial price.

The performance value of a task event is the reward for its performer. The performance u_e is split between the task performer, disputer, and verifier. The performance value's funding sources are staked tokens when the task is created and cost of the task. If there is a dispute, the disputer stakes additional tokens to add to the sum of u_e. Then, if the dispute is verified, the performer receives $u^v_{e,1}$, dispute verifier $u^v_{e,3}$, and task disputer $u^v_{e,4}$, where $u_e = u^v_{e,1}+u^v_{e,3}+u^v_{e,4}$. If the dispute is unverified and expires, $u^d_{e,1} < u_e$, and $u^d_{e,3} = u^d_{e,4} = 0$. A dispute that has been withdrawn is treated the same as no dispute. The task performer gets the entire u_e. Each attestation request has a reward value, v_e. The attestor receives the reward only if their report is approved by the NFT buyer.

3.2 Reputation

The protocol assigns a reputation to underwriters, servicers, and buyers. Each participant's reputation is based on their staked tokens and history of transactions, disputes, and attestations. Buyers refer to the reputation of underwriters and servicers to evaluate the legitimacy of an asset. The reputation of a buyer prevents malicious disputes against underwriters and servicers. The following subsections describe how reputation is computed for various market participants.

3.2.1 Underwriter Reputation

The reputation of an underwriter has five contributing components: transaction history, seller dispute, unverified buyer dispute, verified buyer dispute, and approved attestations.

These components are denoted by $\{1 \cdots 5\}$. Each of the components have a cumulative effect cf. The cumulative effect of the transaction history is cf_1. For each component $s \in \{1 \cdots 5\}$, a scaling function θ_s is used to normalize the relative importance of reputation components. For example, a buyer dispute that is verified by the seller would more negatively impact a seller's reputation than an unverified buyer dispute. Each cumulative effect is normalized by a function ξ, whose growth rate is controlled by γ_s. The adjusted component contribution is $\theta_s \xi_{\gamma_s}(cf_s)$. And, the underwriter's reputation is normalized by κ_c.

Reputation is time bounded. A market participant has nonzero reputation for as long as SPT tokens have been staked. Each staking action updates a participant's reputation, and its expiration time is $RT(\iota)$. Reputation can be updated based on ongoing task events, but staked tokens cannot be withdrawn until the expiration time is reached. We use $R(\iota, t)$ to denote the reputation of agent ι at time t, and $R(\iota, t) = 0$ if $t > RT(\iota)$.

For an asset class c, the cumulative effect of transaction volume is

$$cf_2^{\iota^1}(i, c, t) = \sum_{a \in A(i,c)} R(\iota_a^4, t)\psi(a), \tag{3.3}$$

where $R(\iota_a^4, t)$ is the buyer's reputation a, and $A(i, c)$ is the set of terminated assets of class c whose underwriter is i. An asset is terminated if the termination task has status *closed*. The adjusted price of each asset is scaled by the reputation of the corresponding buyer. A transaction gives the underwriter's reputation a higher boost if the counterparty is a buyer with higher reputation. This also prevents an underwriter from faking transactions to increase reputation. Additionally, an underwriter is prevented from faking transactions by impersonating buyer accounts since buyer accounts must stake tokens to maintain reputation. See 3.3 for a more detail discussion.

The cumulative effect of seller dispute is

$$cf_2^{\iota^1}(i, c, t) = \sum_{a \in A(i,c)} \psi(a) \mathbb{1}_{\{d_{\tilde{e}_a}^{\iota^3}\}}, \tag{3.4}$$

where $\mathbb{1}_{\{d_{\tilde{e}_a}^{\iota^3}\}}$ whether seller has submitted a dispute against the underwriter. The protocol does not assign a reputation to sellers but the seller and underwriter continue to have off-chain interactions. The seller is entitled to dispute the underwriter's statements in the creation task event. This is a summation of the adjusted price of all the assets that are disputed by the seller. Note that the parameter t does not affect the cumulative effect in this component.

An unverified buyer dispute is defiend by: a buyer dispute that is not verified by the corresponding seller, and where the buyer and the underwriter decide not to resolve their difference off-chain, and the buyer does not withdraw the dispute and leaves it to expire. An unverified dispute can penalize both the buyer and underwriter in staked tokens and reputation. The cumulative effect is

$$cf_3^{\iota^1}(i,c,t) = \sum_{a \in A(i,c)} \sum_{e \in E_a^{\iota^1}} R(\iota_a^4, t)\psi(a)(u_e - u_{e,1}^d)\mathbb{1}_{\{d_e^{\iota^4}\}}, \tag{3.5}$$

where $E_a^{\iota^1}$ is the set of tasks whose performer is the underwriter, $u_e - u_{e,1}^d$ is the loss of performance value, and $\mathbb{1}_{\{d_e^{\iota^4}\}}$ is whether this task has an unverified dispute. The adjusted price of an asset is scaled by the reputation of the buyer. This scaling gives more dispute power to the buyer who is better reputed. It also prevents a potential adversary from falsely disputing an underwriter. See 3.3 for more information.

The cumulative effect of verified buyer dispute is

$$cf_4^{\iota^1}(i,c,t) = \sum_{a \in A(i,c)} \sum_{e \in E_a^{\iota^1}} R(\iota_a^4, t)\psi(a)(u^e - u_{e,1}^v)\mathbb{1}_{\{d_e^{\iota^3,\iota^4}\}}, \tag{3.6}$$

where the $\mathbb{1}_{\{d_e^{\iota^3,\iota^4}\}}$ is whether the task event e has a verified dispute. The rewards received by seller ($u_{e,3}^v$) and buyer ($u_{e,4}^v$) are paid by the penalty on the underwriter.

The cumulative effect of approved attestation is

$$cf_5^{\iota^1}(i,c,t) = \sum_{a \in A(i,c)} \sum_{e \in E_a^{\iota^1}} v_e \mathbb{1}_{\{\alpha_e^{\iota^4}\}}, \tag{3.7}$$

where the $\mathbb{1}_{\{\alpha_e^{\iota^4}\}}$ is whether the task has an approved attestation. The value v_e is the reward that the attestor claims. An underwriter gains reputation from creating more incentives for attestations but these come with a cost. A rational underwriter decides on the number of attestations depending on reputation and cost.

The underwriter's reputation in asset class c is the summation of all the cumulative effects.

$$R(i,c,t) = h(i,c,t) \sum_{s \in \{1,2,3,4,5\}} \theta_s \xi_{\gamma_s}(cf_s^{\iota^1}(i,c,t), \tag{3.8}$$

where $h(i,c,t)$ is the amount of staking that underwriter i is committed to asset class c up to time t, θ_s is the component specific normalizing factor, and γ_s is the decay rate. θ_s can

be described as the amplitude of the cumulative effect, and γ_s is how fast the cumulative reaches its maximum potential. Aggregating across asset classes, the total reputation of the underwriter is

$$R(i,t) = \sum_{c \in C(i)} \kappa_c^{\iota_1} R(i,c,t), \tag{3.9}$$

where $C(i)$ is the set of asset classes that underwriter i has underwritten.

3.2.2 Servicer Reputation

The reputation calculation for a servicer is similar to an underwriter. In many cases, the same off-chain entity will play the role of both underwriter and servicer. Regardless, a separate reputation is maintained for each role. The underwriter performs tasks needed to register NFTs to the marketplace, and the servicer facilitates the information and payment exchange between the seller and buyer throughout the life cycle of an asset. The protocol separates both roles to enable some underwriters to pass on the work of servicing to specialists, allowing them to focus on asset origination. The reputation of a servicer depends entirely on the tasks performed.

The cumulative effect of seller dispute is

$$cf_1^{\iota^2}(i,c,t) = \sum_{a \in A(i,c)} \sum_{e \in E_a^{\iota^2}} \psi(a) \mathbb{1}_{\{d_{ea}^{\iota^3}\}}. \tag{3.10}$$

The cumulative effect of unverified disputes,

$$cf_2^{\iota^2}(i,c,t) = \sum_{a \in A(i,c)} \sum_{e \in E_a^{\iota^2}} R(\iota_a^4, t) \psi(a)(u^e - u_{e,1}^d) \mathbb{1}_{\{d_e^{\iota^4}\}}. \tag{3.11}$$

The cumulative effect of verified disputes,

$$cf_3^{\iota^2}(i,c,t) = \sum_{a \in A(i,c)} \sum_{e \in E_a^{\iota^2}} R(\iota_a^4, t) \psi(a)(u^e - u_{e,1}^v) \mathbb{1}_{\{d_e^{\iota^3,\iota^4}\}}, \tag{3.12}$$

The cumulative effect of approved attestations,

$$cf_4^{\iota^2}(i,c,t) = \sum_{a \in A(i,c)} \sum_{e \in E_a^{\iota^2}} v_e \mathbb{1}_{\{\alpha_e^{\iota^4}\}}. \tag{3.13}$$

The servicer's reputation for an asset class is

$$R(i,c,t) = h(i,c,t) \sum_{s\in\{1,2,3,4\}} \theta_s \xi_{\gamma_s}(cf_s^{\iota^2}(i,c,t)). \tag{3.14}$$

The total reputation of buyer i is

$$R(i,t) = \sum_{c\in C(i)} \kappa_c^{\iota 2} R(i,c,t). \tag{3.15}$$

3.2.3 Buyer Reputation

Buyer reputation has four cumulative effects: transaction history, unverified dispute, verified buyer dispute, and attestation approvals. For an asset class c, the cumulative effect is $cf_s(i,c,t)$, for any $s \in \{1 \cdots 4\}$. The cumulative effect of transaction volume is

$$cf_1^{\iota^4}(i,c,t) = \sum_{a\in A(i,c)} \psi(a). \tag{3.16}$$

Note that we use the adjusted price of the assets to measure each asset's contribution. The adjusted price takes asset duration into account. The contribution has a simple interpretation that reputation increases as the buyer participates in more transactions. This cumulative effect does not grow linearly and is transformed by a scaling ξ curve.

The cumulative effect of unverified disputes is

$$cf_2^{\iota^4}(i,c,t) = \sum_{a\in A(i,c)} \sum_{e\in E_a} \psi(a)(u^e - u^d_{e,1}) \mathbb{1}_{\{d_e^{\iota^4}\}}. \tag{3.17}$$

The cumulative effect of verified dispute is

$$cf_3^{\iota^4}(i,c,t) = \sum_{a\in A(i,c)} \sum_{e\in E_a} \psi(a)(u^v_{e,3} + u^v_{e,4}) \mathbb{1}_{\{d_e^{\iota^3,\iota^4}\}}, \tag{3.18}$$

These two cumulative effects are similar, but they cannot be directly compared. They are scaled by a growth function ξ_s, and then multiplied by θ_s.

The cumulative effect of attestation approvals is

$$cf_4^{\iota^4}(i,c,t) = \sum_{a\in A(i,c)} \sum_{e\in E_a} v_e \mathbb{1}_{\{\alpha_e^{\iota^4}\}}, \tag{3.19}$$

The inclusion of this cumulative factor incentivizes the buyer to approve useful attestation reports.

The buyer's reputation in asset c is the summation of the cumulative effects,

$$R(i,c,t) = h(i,c,t) \sum_{s\in\{1,2,3,4\}} \theta_s \xi_{\gamma_s} (cf_s^{\iota^4}(i,c,t). \tag{3.20}$$

The cumulative reputation of the buyer is

$$R(i,t) = \sum_{c\in C(i)} \kappa_c^{\iota_4} R(i,c,t). \tag{3.21}$$

3.2.4 Asset Reputation Liability

The protocol rate-limits the value of assets that can be underwritten. Each asset's weight is measured as a liability against the underwriter's reputation. Not all assets contribute the same amount of risk. For example, an asset has higher risk if it is has a higher value and is contracted on for a larger amount of off-chain assets. A debt asset has more weight if it has a higher principal and has a longer duration. An asset with higher default rate has a higher liability. Reputation helps determine the liability threshold of the underwriter.

The reputation libaility of an asset has three components: exposure risk, the duration risk, and the specific risk of the asset class. The last component describes how likely the creation and termination of assets are disputed. There are three type of dispute factors: seller dispute, unverified buyer dispute, and verified buyer dispute. The first dispute factor is

$$DF_1(c) = \frac{\sum_{a\in A(i,c)} \psi(a) \mathbb{1}_{\{d_{\tilde{e}_a}^{\iota_3}\}}}{\sum_{a\in A(i,c)} \psi(a)}. \tag{3.22}$$

Note that $DF_1(c) \in [0,1]$. It is a weighted probability of how likely an asset would be disputed by the seller. The unverified buyer dispute factor is

$$DF_2(c) = \frac{\sum_{a\in A(c)} \psi(a) \mathbb{1}_{\{d_{\tilde{e}_a}^{\iota_4}\}}}{\sum_{a\in A(c)} \psi(a)}. \tag{3.23}$$

The factor attributing to verified disputes is

$$DF_3(c) = \frac{\sum_{a\in A(c)} \psi(a) \mathbb{1}_{\{d_{\tilde{e}_a}^{\iota_3,\iota_4}\}}}{\sum_{a\in A(c)} \psi(a)}. \tag{3.24}$$

The total dispute factor is $DF(c) = DF_1(c) + DF_2(c) + DF_3(c)$. Let c_a be the asset class of a. The asset risk of a is

$$r(a) = \beta_a \psi(a) \left[\rho^l_{c_a} + \rho^m_{c_a} \int_0^T \frac{1}{(1+\tau_{c_a})^t} dt + \rho^n_{c_a} DF(c_a) \right], \tag{3.25}$$

where $\rho^l_{c_a}, \rho^m_{c_a,2}, \rho^n_{c_a}$ are the scaling factors for the exposure risk, duration risk, and specific risk component. For underwriter i, The condition for reputation liability is

$$\sum_{a \in A(i,c)} r(a) \leq R(i,c,t). \tag{3.26}$$

3.3 Incentives

The mechanism design of staking, attestation, dispute, and reputation creates incentives for market participants to act truthfully and guard against a counterparties' dishonest behavior. In this subsection we discuss how the protocol's mechanism design impels participants to act to optimize market outcomes.

3.3.1 Asset Administrators: Underwriter and Servicer

Asset administrators work directly with sellers to underwrite and service assets. An administrator has an information advantage since sellers rely on administrators to represent and value their assets truthfully. The data pertaining to an asset is stored on-chain, but interactions between seller and administrators could still involve the exchange of information off-chain, contributing to assymetric power dynamics.[2] An underwriter, for instance, may falsely undervalue an asset while claiming otherwise, or a servicer may dishonestly underreport the operating income of an asset.

To disincentivize dishonest behavior, the protocol specifies a dispute mechanism. For example, both the seller and servicer know how much income is collected and paid, but a buyer only sees what the servicer reports on-chain. The buyer can challenge the servicer by offering the seller a reward in exchange for verifying whether the servicer cheated in the on-chain payment distribution. This dispute mechanism serves a means for sellers and buyers to keep an asset administrator in check.

Underwriters are incentivized to register legitimate assets that have interested buy-

[2]See section 5.3 for a detailed discussion of data privacy and security.

ers. Their reputation is negatively impacted by disputes. To compensate for reputation, an underwriter stakes additional tokens or requests attestations to provide greater legitimacy to the assets they offer.

The reputation liability condition (see equation 3.26) limits the number of assets an underwriter can underwrite. A new underwriter is required to stake tokens to establish reputation and can use them as insurance in case of disputes. The protocol incentivizes an underwriter to build on an established reputation rather than register as a new entity.

Unwarranted disputes are mitigated as both the administrator and the buyer incur a reputation and staking penalty for unverified disputes. In most cases, the buyer disputes only if he believes that the seller is likely to cooperate to verify that the administrator fails to perform appropriately. The buyer has the option to dispute unilaterally to punish the administrator (see equation 3.5 and 3.11). While this is not an optimal action in a single-shot stage game, this punishment is a credible threat if the buyer and asset administrator have repeated interactions with one another.

An asset administrator would also want to guard against a bad actor from acquiring assets solely to dispute and inflict reputational damage. The mechanism prevents anonymous agents from using the dispute system to undermine legitimate market participants' reputation because only the reputational damage of a dispute is proportionate to the disputing buyer's reputation. To perform the attack, the bad actors have to stake tokens to maintain a reputation. Furthermore, the bad actors pay tokens as a penalty for each unverified dispute, making such an attack cost prohibitive.

3.3.2 Seller

A major concern for a seller is whether the underwriter structures a fair contract. The seller trusts the underwriter to provide accurate information about an NFT and the seller negotiates with the underwriter to publish all data and the methodology used in the underwriting process. The data is protected by a privacy and security guarantee. If the underwriter publishes false information about the NFT, the seller can unilaterally dispute.

An additional concern for a seller is if the servicer misreports the the seller's contractual performance. For example, the seller finances a development project by issuing debt as NFTs. The project generates a healthy operating income on behalf of the NFT's investors. A servicer may take a larger cut of the monthly income than what is agreed on. Since the seller has access to on-chain payment information, the seller can submit a dispute, challenging the

servicer.

Alternatively, a seller can falsely submit a dispute against an asset administrator. This case is similar to a consumer who has the right to file complaints against any business to the Consumer Protection Agency. Sellers can only initiate a dispute with an underwriter they have transacted with. False disputes are disincentivized between sellers and asset administrators that trade with each other more than once.

3.3.3 Buyer

Buyers use Spot primarily to assess the validity of assets underwritten on-chain. In equilibrium, the buyer relies on the underwriter's reputation to validate the information of an asset. If the reputation of the underwriter alone is insufficient, the buyer uses attestations to confirm the validity of the information provided by the underwriter.

A buyer uses the protocol's dispute mechanism to keep the asset administrators in check. If there is suspicion of dishonest statements by administrators, the buyer can dispute and offer a reward to the seller to check if the administrators have cheated. Upon seeing this public dispute, the asset administrator would be likely to either negotiate with the buyer off-chain or risk the seller verifying the dispute. Thus, the administrator's optimal action is to ensure the validity of asset information for buyers to make informed purchasing decisions.

Another case for potential fraud against the buyer is when an adminstrator colludes with the seller. This type of collusion is no more dangerous than an independently fraudulent underwriter. Instead of colluding with a seller, an underwriter can create a false seller account. The underwriter does not gain from underwriting bad NFTs because unverified buyer disputes have negative repercussions. The reason an underwriter would not underwrite fraudulent NFTs is the same as what precludes underwriter and seller collusion.

4. Summary

Spot aims to establish a trustful marketplace for NFT assets. The protocol establishes a set of rules for how market participants register, service, and buy and sell those assets. Mechanisms for attestation, dispute, staking, and reputation system allow buyers to evaluate assets that are not backed by a centralized authority.

A mature decentralized market for NFTs requires a set of well-defined, common core standards. At a minimum, there needs to be a protocol for underwriting and servicing these

NFTs, a NFT wallet integrating with assets' lifecycle, and an exchange for trade. Spot tackles the underwriting and servicing component.

5. Appendix

The description of the protocol has been kept generic. The protocol could be implemented on any public blockchain that allows programmable smart contracts. Our target blockchain is Ethereum, but the protocol can also be implemented on other blockchains such as NEO or EOS. We discuss Ethereum-specific implementation details in the appendix.

5.1 ERC-721: Tokens Permission and Exchange

The assets in the Spot Protocol are represented by tokens compliant to the ERC-721 standard. Each ERC-721 token is unique. For example, buying a token is equivalent of buying the financial interest in a home mortgage. The token holder is entitled to a monthly payment for 2 years, and the repayment of the principal. The ERC-721 standard defines a common interface that allow the tokens to be managed and traded. Other projects are starting to adopt the same standard.

The project 0x supports decentralized atomic swap of ERC-721 tokens. 0x is a building block of a decentralized exchange of Ethereum-based tokens. 0x enables the trade of ERC-20 tokens without a centralized authority. By adopting the same non-fungible token standard as 0x, any project that already employs 0x is able to seamless integrate with the tokens issued by Spot.

The R-token standard extends to ERC-20 standard to define a mechanism to control how tokens can be compliantly transfer (Gavin, 2018). Spot's SPT tokens follow the ERC-721 standard. The extensions as proposed by R-token to ERC-721 tokens can be applied on SPT. We call that permissioned ERC-721. These permissioned tokens check the on-chain Regulatory Service to ensure compliance with Know Your Customer (KYC), Anti-Money Laundering (AML), tax and other considerations.

5.2 Verification Game and Reputation Score

The Ethereum EVM has strict restrictions on what is available for computation and within the gas limit of a block. The reputation scoring system, such as 3.4, **??**, and 3.6, relies on using extensive transaction and event history. Unless we store all of the transaction data and the entire history of task events on smart contract storage memory, the EVM does not have access to all the relevant information. Storage costs would be prohibitive and the block gas limit would preclude complex computations.

TrueBit proposes a solution for executing complex computation problems on the blockchain. It is possible to implement TrueBit (Teutsch and Reitwießner, 2017) to offload the computation off-chain, and it only requires on-chain calculation when there are disagreements about its results. TrueBit's scalable computation system is secured and trustful. The downside of using TrueBit is that the system is not fully ready, and even a simplified, customized implementation requires that a sufficient number of verifiers exist in the network.

Instead, we propose a relatively straightforward verification game to computer reputation scores. The agent's reputation $R(i,t)$ is tied to the staked token amount $h(\iota,c,t)$. The agent's reputation is computed off-chain and issues a smart contract-based request for an update. In that update request, the agent verifies n previous requests submitted by others. A request is accepted only if it has received n verifications whose majority decision concurs with the request. The verifiers that supports the majority decision get a reward given by the reputation submitter. If verifications do not recommend an update, the request submitter is penalized. The verifiers that support the minority decision is also penalized. This system is similar to Augur (Peterson, Krug, Zoltu, Williams, and Alexander, 2018).

5.3 Decentralized Data Privacy and Security

The protocol uses an **EntityRegistry** to uniquely identify each market participant with a private key, a 32 byte long number. We use the same elliptical curve (secp256k1) to generate a public address as in (Gavin, 2018). The entity owner is required to provided identity information to participate in the marketplace. The identity information is stored in an encrypted, privacy controlled data object.

Property	**Solidity Type**	**Note**
`publicAddress`	`address`	
`dataObjectHash`	`bytes32`	Decentralized data object link

Table 1: Smart Contract Data Field for Identity Registry

Each data object stored in the decentralized data storage follows a specification that allows the data to have varying levels of privacy. It could be private or public key encrypted, or public. The highest level of privacy is privately encrypted data. The owner of privately encrypted data is hidden from view. Data objects that are encrypted by public key are viewable only to the data owner, but ownership information itself is public. Public data is

viewable to all.

The data object is addressable to a hash. Each object generates a hash. The hash is globally unique, immutable identifier. If any of its content is changed, it becomes an entirely different object. We uses proto3 to encode data. The content of an object is an serialized protobuf. Every data object follows the same upper level protobuf definition. The upper level definition describes the privacy level, data definition, and a data field. The encrypted data is stored in the data field.

References

Gavin, W. (2018): "Ethereum: A Secure Decentralised Generalised Transaction Ledger," Byzantium Version, http://gavwood.com/paper.pdf, Accessed: 2018-01-13.

Gupta, V. (2018): "Mattereum: Smart Contracts, Real Property," https://mattereum.com/images/pdf/mattereum-draft-white-paper.pdf, Accessed: 2018-01-13.

Jackson, M. O. (2003): "Mechanism Theory," *Optimization and Operations Research*, III.

Mazieres, D. (2018): "The Stellar Consensus Protocol:A Federated Model for Internet-level Consensus," https://www.stellar.org/papers/stellar-consensus-protocol.pdf, Accessed: 2018-01-13.

Peterson, J., J. Krug, M. Zoltu, A. K. Williams, and S. Alexander (2018): "Augur: a Decentralized Oracle and Prediction Market Platform," https://www.augur.net/whitepaper.pdf, Accessed: 2018-01-13.

Remeika, B., A. Amano, and D. Sacks (2018): "The Regulated Token (R-Token) Standard," Version 1.0.0, https://harbor.com/rtokenwhitepaper.pdf, Accessed: 2018-01-13.

Teutsch, J., and C. Reitwiessner (2017): "A scalable verification solution for block," https://people.cs.uchicago.edu/ teutsch/papers/truebit.pdf, Accessed: 2018-01-13.