2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020887

Rental NFT Protocol With Advanced Rewards Splitting

Andrew Kirillov
BS, M.E.T.
UC Berkeley 2022

Abrar Rahman
BA, Computer Science
UC Berkeley 2023

Ayush Aggarwal
BS, EECS
UC Berkeley 2022

Abstract

Multiple approaches to NFT rentals have surfaced this year as blockchain games continue to proliferate but a standard solution has not been adopted, leaving the game ecosystems independent of each other. This paper defines an auction-based system to decouple usage and ownership of a playable NFT, which functionally creates a decentralized mechanism to carry out scholarship for GameFi economies. Our system can be viewed as a proposal for standard interfaces between games, making rental a primitive instead of a product. [1]

1 Problem Definition

1.1 Motivation

NFT play to earn economies lack standard decentralized methods of scholarship, which is the phenomenon of NFT owners loaning out playable game assets to a player-for-hire in exchange for a ratio of the liquid assets earned from gameplay. These reward splits between owner and player, also known as manager and scholar [1], are currently negotiated over message and the rental occurs through exploiting a centralized front-end system. For this reason, on-chain NFT rental mechanisms are ill-defined. There has not been significant time to adopt protocol standards let alone communal norms surrounding NFT rental.

Out of this uncertainty, some projects have established the need for scholarship mechanisms. This will be covered with greater depth in the literature review, uncovering the need for decoupling of utility and ownership for all NFTs. Market dynamics which drive the rest of the financial ecosystem in blockchain protocols seem to not be established when it comes to NFT assets. Thus, it is extremely important for the sake of games, art, and new types of assets to be represented on blockchains with the same access to financial primitives which bring liquidity and utility. The largest driver of users will is the ability to make money in new ways, which has been recorded for DeFi in summer of 2020 [2]. For illiquid assets, adding utility through standardized protocols could have a similar impact on many other verticals of life.

The key contributions of this paper come from synthesizing research of existing methods and understanding needs of GameFi projects in today's ecosystem. We propose an extension of existing ERC standards and form a proposal aiming to move the current off-chain rental norm on-chain. This could potentially have enormous benefits to cryptocurrency utility. Blockchain games would be more inter-operable because the assets driving them could be composably rented to create financial connections between disparate game ecosystems. Additionally, there has been lots of rugpulls and scams which stem from speculation [3]. Artists and creatives who use NFTs currently would have an easier time understanding the real currency value of their work, and more liquidity in the NFT space could mean less room for rugpulls and scams, as the effects speculation would even more quickly play out.

1.2 Contributions

In our research, we have discovered an interesting insight: that none of the current implementations leverage a mechanism by which the distribution of rewards between ”scholar” and ”manager” can be determined by the market, as opposed to the manager. Part of the principles of decentralized technology is that markets are good at understanding the natural balance point of things. Scholars need a formal way to ensure the rates they are getting are fair, since as of now the negotiation is facilitated and backed by social consensus of good rates via Discord channels and online forums.

The basic way of achieving a fair market value is competitive bidding through an auction [4]. We pro-

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

pose that market defined rental terms will be much better for the ecosystem than listing prices without negotiation. In order to package this idea into a usable concept for Ethereum developers, the sample impelmetation extends the basic ERC721 as well as a popular proposal ERC4907. This functions as an updated EIP to be considered by the community.

2 Review of Existing Protocols

2.1 Axie Infinity

Axie Infinity is, indisputably, the most popular blockchain game. Originally released in 2018, it now boasts almost 3 million daily active players, with over $3.6B in volume traded in its marketplace and over $1B held in its community treasury [1].

Similarly to CryptoKitties, Axie Infinity is centered around collectible creatures (Axies) that display traits of varying rarities, that can be bred. However, Axie is more developed along all three pillars of gameplay, economy, and infrastructure.

Off-chain systems are important now to the development of player ecosystems for various games. Axie Infinity, for example, has a robust scholarship ecosystem based around Discord channel and forum discussion. We were able to attend an Axie Infinity event at GDC, which exposed us to many founders working on this problem space. It is clear that in the community, Axie Infinity signals leadership. Working with the engineers and leaders of this game and the broader YGG guild is necessary to undersatnd the full picture.

While Axie Infinity is a game that is also backed by the Ethereum blockchain, the Axie Infinity team (Sky Mavis) actually developed an Ethereum sidechain called Ronin on which all Axie Infinity assets are stored. This brings scale to Axie in terms of increased transactional throughput, and in terms of decreased gas cost - an infrastructure development that is rare to see in other blockchain games as of yet. This makes the game more accessible to newer players (to some extent, as the assets themselves have become fairly expensive), and opens up doors for future avenues of innovation, unhindered by the lack of scale on the Ethereum main chain. This exemplifies a bottleneck when designing a solution for NFT rental: most games have proprietary technology that requires individual integration over time.

2.2 Double Protocol

Double Protocol is a fully decentralized NFT rental protocol aiming to connect the NFT owners with users in an on-chain marketplace. The project comes complete with a sublet option and reservation system. The benefit of Double Protocol is that they are compatible out-of-box with ERC721 tokens, and have integration with Decentraland and Warrina already. On-chain marketplace for rentals is a common product idea emerging recently, and Double Protocol is an example of what this type of solution looks like.

Something key to note is that while solutions can be decentralized on-chain, if integrations are required it weighs heavy responsibility on the developers to continue maintenance and integrations as more metaverse applications arrive. Without a solid roadmap to decentralize the responsibility, this functions similarly to a true centralized solution without the potential cost efficiency benefits of a third party.

Notably, they also briefly flashed an Ethereum improvement proposal with the goal of decoupling use and ownership of an NFT. The essential quality of lending for GameFi is this decoupling of an NFT to its owner. If written into protocol, this opens up a wide array of use cases that will drive liquidity, and users, into the GameFi space.

2.3 BreederDAO

BreederDAO is the go-to asset production factory for quality gaming assets of the Metaverse. They generate curated assets that deliver best value at scale for some of the largest guilds in the Metaverse so they can supercharge their play-to-earn economies. All asset generation tools are developed in-house and derived from a combination of comprehensive game mastery, data-driven analytics and modeling. BreederDAO is committed to equipping the next billion players and becoming a pillar of the play-to-earn economy [5].

Notably, we were able to talk to the founder of BreederDAO at the Axie Infinity event in San Fransisco during GDC. This event made it possibly to get real feedback on our research from notable people in the space. The founder spoke about how they are actively investing in NFT rental projects, which all take the same basic form - Owner transfers to a proxy contract, while user enjoys the benefits through this extended system.

From notes during that conversation, it was brought up that ”liquidity results in attracting users, and renting an NFT takes it away from the liquidity pool”. This is important since it underpins why most of the current implementations haven’t established dominance over the market. To really solve this liquidity problem, the solution must keep the NFT in the liquidity pool. Phrased differently, only selling an NFT should move it from one liquidity source to another.

2.4 Aavegotchi

Aavegotchi is a crypto collectible game built on Ethereum where participants can purchase and grow Aavegotchis, NFT avatars used to explore and interact with Aavegotchi's digital universe. Each Aavegotchi avatar is a digital collectible that has unique attributes defined by a rarity score. The game has its own rental mechanism built already, and is open-source.

The lending is implemented via two doubly linked lists that list all the open rentals and all the finalized agreements. These are stored in AppStorage and function as the main record keeping system for rentals. Other protocols that generalize to multiple games essentially use this same idea but in a separate contract and must integrate with individual game contracts and asset interfaces.

Aavegotchi is the best example of how a rental protocol would work in a siloed environment (i.e. all games did rentals individually within their own smart contract systems). In our research, the only thing most folks agreed upon with regards to this topic was that the Aavegotchi model is the baseline of our best working solution for NFT rentals. In order to extend this to art, other games, and more it will require a deeper change in the Ethereum blockchain through an EIP. Ideally, this change would take the Aavegotchi style approach and make it part of the NFT interface natively [6].

2.5 ERC4907

ERC4907 standard is an extension of ERC-721. It proposes an additional role (user) which can be granted to addresses, and a time where the role is automatically revoked (expires). The user role represents permission to "use" the NFT, but not the ability to transfer it or set users. This is the key innovation that would make NFT marketplaces more attractive to developers of blockchain games. The underlying baseplate protocol of lending is necessary and likely will require a change to the core Ethereum protocol such as this EIP [7].

Thus, many projects have written about or used the ERC4907 in some capacity. Its main value is distinctly separating usage and ownership at the smart contract level, similar to how Aavegatchi chose to handle their rental feature internally [8]. For the propsoal we came to, it made sense to build an auction system on top of this ERC.

3 Approach

We designed and implemented a new ERC standard extending ERC721 and ERC4907, which boasts an auction system. If adopted, this would allow NFTs to be lent and borrowed native in smart contracts and through user transactions. More work is there to be done in integration with current efforts of the Ethereum foundation as well as methodology for designing token standards. Here, we present the design and implementation details that drove us to our contract attached in the appendix.

3.1 Design

- **Design goal 1:** We aimed to separate notions of usage and ownership, to maintain the liquidity benefits of owning an NFT of a certain ecosystem. From our learning that projects in industry want a protocol that can allow an owner to list an asset for sale at the same time as renting it out, we understood that sale and use must be decoupled to allow for this.

  By extending the ERC4907 specification, we were able to use the key functions surrounding a new "user" field in the NFT. This is different from the standard ERC721, which places utility functions and transfer functions on the same privilege level. By splitting privileges between ownership transfer and use, we can create clear delineations for the auction system.

- **Design goal 2:** We aimed to create a system which has market defined reward splits between owners and renters, to support the notion that fair value is derived from competitive markets. The crux of our contribution lies here, in making the scholarship mechanisms of blockchain games more formal and market driven. This is the paradigm that may bring liquidity and users to the NFT space, according to our interveiws with project founders in the GameFi space.

  We were able to execute on this by implementing an auction mechanism where potential renters outbid each other by accepting smaller cuts of the rewards. This is important because it resembles how the negotiations were already happening on Discord from our observations. We surveyed multiple communication hotspots where scholars and managers gave out rents and found that values too unfavorable for the renter were socially rejected through negative reactions on discord and negative comments. Thus, we decided that an auction would turn this into a more precise market, giving both sides better clarity into the real value of scholarship.

- **Design goal 3:** We aimed to hit a sweet spot in market fit, where our product can easily generalize to NFT games, but still cover all the bases that this unique use case needs. A broad solution would miss edge cases, and a hyperspecific solution would sacrifice interoperability between games.

By allowing for owner/user splits on any number of reward tokens per rental, we essentially created an interface which is modular. For example, Axie Infinity rewards users through allocation of AXS and SLP tokens which have different utility. The smart contract system we built already defines an arbitrary number of tokens relating to reward splits, so an integration would be simple even if they decide to add token types in the future. As long as the tokens are granted to the NFT holder, any number could theoretically be supported by our system

- **Design goal 4:** We aimed to use the the best Solidity design patterns and maintain the standards set by Ethereum foundation and broader community members over time. This is important from a research perspective to understand how standards evolve and how a new idea can manifest itself.

  The key discussion on this resolved by extending finalized ERCs, rather than re-writing similar code. This allows the community to understand it better given they have an existing grasp on the current methodologies for ERC721. We also made sure to optimize gas costs via minimizing internal function calls, use struct packing, uint sizing, proper state management, implementing basic & efficient data structures (i.e. linked list), proper security assertions, and event emission.

3.2 Implementation

- **Contract Definition:** The contract contains 3 data structures, which support the relevant fields to a rental covering all bases of an owner and user. There are structs for Rental, Listing, and a Bid. Each struct has a mapping from address to uint8 recording the ratio of reward tokens to reward amount kept by the borrower. 5 external functions (postListing, cancelListing, placeBid, cancelBid, acceptBid) will be used to conduct the English auction. A basic workflow for this would be the following:

  1. Owner calls postListing with initial splits and auction duration
  2. Potential Users call placeBid until the duration ends or Owner is satisfied with bid
  3. Owner calls acceptBid which creates a rental
  4. a tokenID is mapped to a rental agreement struct until clawback occurs

  4 internal functions (_removeListing, _removeBid, _isBidBetter, _beforeTokenTransfer) will be used to manage permission overrides and interface with contract data structures privately to minimize attack surface.

- **Flexible Extension:** In implementing the _isBidBetter function, we made this a virtual function (i.e. can be overwritten by a contract that inherits this) to allow others to define their own logic for what makes one bid better than another. Especially in cases where there are multiple reward tokens, there could be a variety of optimizer functions that an Owner would use to understand the best bid. Otherwise, it'd just be a simple check of if the bid being placed has a smaller reward split for the user than the previous one. We provided a naive implementation that checks that all splits for all reward tokens are $\leq$ those of the previous bid, and that the reward split for at least one token is $<$ than in the previous bid.

- **Liquidity Protection:** We overrode the ERC4907 implementation of the _beforeTokenTransfer function to not unset the user of the NFT upon transfer to allow for rentals to persist between ownership changing hands. This is critical to the project, and is a feature that slightly changes the paradigm of how rentals work today. If ownership can change hands while rental is executed, it truly decouples the notion of an owner and user.

- **Bid History:** In order to implement the ability to cancel bids, we had to preserve the entire history of bidding. Without this, all we need to do is keep track of the latest, most competitive bid. But in order to enable this, we implemented an efficient linked list structure composed of two mappings, one storing the head of the LL for a given auction, and one storing the LL node ids to allow for indexing / traversal. This was interesting to build without being able to use pointer variables. This also draws back to the efficiency optimizations and trade-offs we had to make when designing this system.

- **Reward Tokens:** In supporting multiple reward tokens, this forced us into setting a caller convention of passing in either initial or offered borrower splits during listing and bidding, respectively, as an array whose size + order mirrors the array of reward tokens stored on the listing. This is an important constraint that must be adhered to by anyone integrating our smart contract into theirs / calling functions on it.

4 Evaluation

The success of the project was measured off qualitative goals rather than quantifiable metrics. We decided to lean into this due to the creative nature of designing a system that must work for various parties and user groups. We came up with a set of 3 guiding metrics that can be discussed.

The first goal was to make a design that is industry oriented, and comes off as a strong solution to the community. In striving to achieve this goal we talked to many stakeholders in GameFi to understand the problems they are trying to solve with NFT rentals. Successfully, we were able to integrate this feedback deeply into the design for our product and create something useful immediately for the people we talked to. The place for improvement is working with the Ethereum community. After we designed our dream solution, we had to deep dive into Solidity to learn how to implement it. Then, we realized that the evolution of standards and interfaces takes a lot of debate and careful nitpicking. This is a feature, not a bug, and we were able to carefully navigate this to make something in the realm of a compromise between the industry side and academic side. For the future, it would be necessary to lean into the core development team and understand the paradigm of Ethereum governance, to see how EIPs are handled in relation to popular support.

The second goal we made for ourselves was to create a novel solution that is different from existing solutions in the market. We analyzed contract code for a dozen implementations and found common traits between them. We decided to make an auction to highlight the ability auctions to create fair market price action. While we set out to make something new, we ran into bottlenecks from Ethereum similar to the other projects. This forced us to creatively innovate and ultimately the system is novel insofar as it extends an existing proposal in a unique way.

Finally, the system is a stable implementation. We learned how to program in Solidity from scratch and were able to pick up on best practices from analyzing contracts of dozens of implementations for rentals. This is key to evaluating success of the project, and significant implementation took the form of more than just lines of code. The journey from novice Solidity programmer to the final status of the project displays a deeply impactful and well researched system.

5 Conclusion and Future Work

NFT rentals are a too-be-perfected primitive of the metaverse. From this project we found that NFT rentals are surprisingly unexplored, since the status quo is working for most people. There is a chicken and the egg problem when it comes to how these standards will evolve in the future. The industry wants things a certain way but the core developers will not change things around until there is a critical mass. However, that critical mass will build slowly if no standardized protocol can solve NFT rentals at the application layer.

Another takeaway is that people are okay with hybrid systems which include centralized components. Axie Infinity scholarship remains a powerful force in blockchain gaming despite the way it manifests in practice. Many users just see the interface and don't really need to understand what is going on underneath. On-chain, off-chain, if they can play they are happy and the economics of the game will persist. Only those thinking about the total system economics beyond individual projects will care about updating standards.

Finally, any new project that wants to use NFTs with utility ought to think about how they are used in relation to standards and other games. They must survey the financial tools that people are using and craft a product that fits within the system or expands it in some way. Overall, the industry is marching towards the right intentions and more innovations will tell us if liquid NFT markets are the choice that the community makes.

5.1 Acknowledgements

The last two months working with Abner as a part of the Berkeley Center for Decentralized Intelligence proved extremely insightful in marching towards a generalized NFT lending protocol design. With the goal of publishing a paper, we have made significant progress in preliminary research and narrowing down to the exact bottleneck we wish to solve. Thank you to Prof. Song for the interest in working with us for a GameFi project and to Abner for the countless connections and ideas that helped us get to a final result.

5.2 Future Work

As mentioned earlier, we must design a leasing protocol that is generalized and works for all types of NFTs. We will be fighting against the physical limitations of the EVM in order to create a novel system that keeps track of rented assets over long periods of time will splitting the ownership and utility factors through a series of smart contracts. For this design to be suggested to conferences and journals, there must be significant research into security surrounding the decoupling of Owner and User. In addition, the rental mechanism at the protocol level likely won't be an auction, and there is work to be done in separating this paper into application and system levels. The finished full-stack design could take a similar role to large DeFi protocols which focus on lending fungible assets.

6 Bibliography

References

[1] AxieInfinity, “Axie infinity whitepaper,” Nov. 2021.

[2] chainalysis, “Part two: The defi explosion explained,”

[3] coindesk

[4] Mccurdy, “The benefits of an auction,” 2020.

[5] BreederDAO.

[6] Aavegotchi, “Massively scaling play-to-earn with gotchi lending,” Nov. 2021.

[7] S. s. Anders (@0xanders), Lance (@LanceSnow), “Eip-4907: Rental nft, erc-721 user and expires extension,” *Ethereum Improvement Proposals*, no. 4907, 2022.

[8] Aavegotchi, “Aavegotchi smart contracts,” 2022.

7 Appendix

Appendix 1: NFT Rental Auction Contract