

# ZerpCraft: A Minecraft Tokenization Project Built On The XRPL

---

Jackson Sardello  
zerpcraftmc@gmail.com



---

**Abstract:**

**While several platforms exist to distribute video game mods, content creators struggle to monetize their creations and are hampered by high middle-man fees. Players do not have an ownership stake in the virtual goods they buy. In this work, I present a novel tokenization method by bridging Minecraft multiplayer servers with the XRP Ledger (XRPL). This paper shows that the tokenization of digital land can be a fast, low-cost, and powerful tool for both server owners and players. The presentation of this solution is the ZerpCraft software platform.**

## Contents

### 1. Introduction

### 2. Architecture

#### 2.1. Zerpcraft Bukkit Plugin

#### 2.2. XUMM

#### 2.3. ZerpCraft NFTs

#### 2.4. NFT Standard: XLS-19 & XLS-20d

### 3. Roadmap

### 4. Acknowledgements

## 1. Introduction

The recent popularity explosion in the NFT market has produced an intense focus on how they might be applied to video games. The appeal of digital ownership is obvious to players who invest large amount of time and effort in virtual worlds. Likewise, publishers and game designers are rushing to create NFT solutions to capitalize on a new wave of gaming monetization.

This advancement is not without its own problems. Most “bottom up” approaches built on-chain are not actually fun. The “top down” approach employed by AAA publishers is seen as a tone-deaf cash grab. Such efforts are facing increased amounts of gamer backlash.

ZerpCraft is an NFT software platform built for content creators. The goal is to give modders, artists, and server administrators a flexible toolset to monetize their work in a way that is still fun for the player. While the top .1% of video game content creators may be able to successfully monetize their work via Patreon or Twitch, most creators work on passion projects with little reward.

---

ZerpCraft allows Minecraft Java Edition server administrators to tokenize their multiplayer worlds into plots of land registered on the XRPL. These plots of land are represented as NFTs that can be bought and sold on-ledger. Server operators can create revenue both through the initial sale of plots of land, but also with ongoing royalties as their NFTs are bought and sold. See the XLS-19 and 20 Architecture section for a more complete breakdown of what capabilities will be available at launch.

The software bridge that is formed between the XRPL and Minecraft is called the ZerpCraft plugin. This plugin is an evolution of the Minecraft XRP Tipper. The plugin is intended to be fully open sourced and free to download by the community after an initial Beta program. Any Minecraft multiplayer server admin will be able to implement the plugin without any up-front costs or special permission.

ZerpCraft NFTs will enforce “property rights” in game by preventing anyone but the NFT owner from building or destroying blocks within the coordinates specified on the NFT. Future improvements are possible including allowing the owner to delegate these permissions to other players while still maintaining the ownership of the NFT.

The XRPL was chosen as the implementation blockchain for a few key reasons.

1. **Cost:** The low fee design of the XRPL gives server admins the ability to issue NFTs for extremely low cost. Likewise the barrier for entry to players should be low. Gas fees or

other ‘miner extracted value’ are not present.

2. **Speed:** The XRPL’s near real-time settlement allows for frictionless interactions between game and blockchain. Game mechanics do not need to be built around long block times keeping the player engaged in the game experience.
3. **Lightweight Design:** The design of ZerpCraft NFTs does not need a heavy smart contract platform implementation. The XRPL’s implementation of NFTs are lean and lightweight, consistent with the rest of the ledger’s design. While other NFT solutions try and shove as much as possible into smart contracts, the Ledger handles the NFT specifics and leaves the game mechanics to the plugin developer.

ZerpCraft is **not** an attempt to create a full blockchain enabled economy in Minecraft. Enabling real-money incentives are compelling, but often take away from the game’s enjoyment. Blockchain + NFT functions should be designed to be mutually beneficial for both the player and the server operator. Play-to-win mechanics will be avoided.

## 2. Architecture

### Zerpcraft Bukkit Plugin

Bukkit is a server-side plugin framework developed for the Minecraft Java Edition multiplayer ecosystem. Utilized by projects such as Spigot and PaperMC, Bukkit plugins are easy to drop into new or existing servers to alter game functionality. The ZerpCraft plugin

---

handles the bridge between NFTs on the XRPL and the Minecraft world. The plugin regularly polls the XRPL to keep track of ownership changes and then enforces the ownership rights based on the information contained in the NFT. The plugin also natively integrates with the XUMM wallet (detailed below) to facilitate low friction wallet interactions. Players simply register their wallet and then the plugin determines if they own any NFT land on the server and assigns the rights to them.

Bukkit plugins are highly flexible and the base ZerpCraft plugin could be enhanced in several ways including:

- In game auctions.
- P2P buying and selling
- Geographic access based on NFT ownership.

Behind the scenes, ZerpCraft utilizes the World Guard plugin, a time-tested project widely utilized by the multiplayer community. World Guard 'Regions' provide the property rights so players can be confident that their land is secured by a best-in-class implementation.

After an initial Beta testing period, the ZerpCraft plugin will be made available for download on plugin hosting websites like SpigotMC and the source code will be available on Github.

## **XUMM**

ZerpCraft is fully integrated with the XUMM wallet. Users register their wallet by performing the `/zcregister` command in-game, scanning a ZerpCraft generated QR code, and

signing an initial "sign in" request. XUMM will also facilitate stamping the NFTs on the Domain for the XLS-19 standard as well as the transition to XLS-20d.

## **ZerpCraft NFTs**

Unlike many NFT projects, ZerpCraft NFTs can be stored completely on-chain and do not need off-chain storage such as IPFS. In the case of property rights, the NFT is a hash containing information that the ZerpCraft plugin needs to perform game-side changes. To prevent duplication of an NFT on a different wallet, the public address will be encoded with the ZerpCraft data. The public address in the NFT stamp must match the public address of the wallet. A decoded ZerpCraft NFT will contain a 3 character prefix for special customizations, a number denoting the issuance number for that particular server, the server's identification code, and a set of coordinates specifying what area of land the NFT represents.

## **NFT Standard: XLS-19**

This standard, proposed by Github user calvincs is intended as a bridge to get ZerpCraft off the ground while the XLS-20d standard is still in development. It is not meant to be a permanent implementation and should be deprecated before ZerpCraft is open-sourced. Having said that, the XLS-19 standard presents an interesting NFT solution that will be perfectly sufficient for launching the product.

XLS-19 may be the first "wallet as an NFT" solution. The standard involves stamping information to the XRP wallet 'Domain' field. Once the user has paid for an NFT, they will

---

receive a push notification from the XUMM wallet asking to stamp the NFT on their wallet Domain. Once the push request is accepted, the ZerpCraft plugin will automatically start enforcing property rights based on the data stamped in the Domain field.

The Domain field contains a hash of the NFT information. The NFT hash is standardized to include some codified data described in the ZerpCraft NFTs section.

**Pros:**

- Wallet based NFTs avoids trustline spam.
- XLS-19 avoids the UX inconsistency of XLS-14. The standard doesn't rely on 3<sup>rd</sup> parties implementing correct display formats.
- Launching on XLS-19 incentivizes purchases from actual Minecraft players and community members instead of speculators. Players will be able to immediately enjoy the property rights given by the NFT, while speculators will be waiting until the launch of the XLS-20d.

**Cons:**

- The XRP wallet Domain field can be overwritten. Because of this, the standard is not optimal for long-term support.
- Moving the NFT via transferal of the Regular Key also transfers any XRP in that wallet, including the reserve.
- No build in DEX support. Buying and selling XLS-19 NFTs without a

marketplace supporting the transaction will be difficult.

- User's will need to complete a migration process to XLS-20d when the standard becomes available.
- The domain field is limited to 256 bytes and thus multiple NFTs are possible, but limited to a small number.
- There is no standard for a single wallet to support multiple XLS-19 projects.

**NFT Standard: XLS-20d**

The XLS-20d standard is a first-class NFT implementation on the XRPL. For a full writeup on the standard, please review the proposal here:

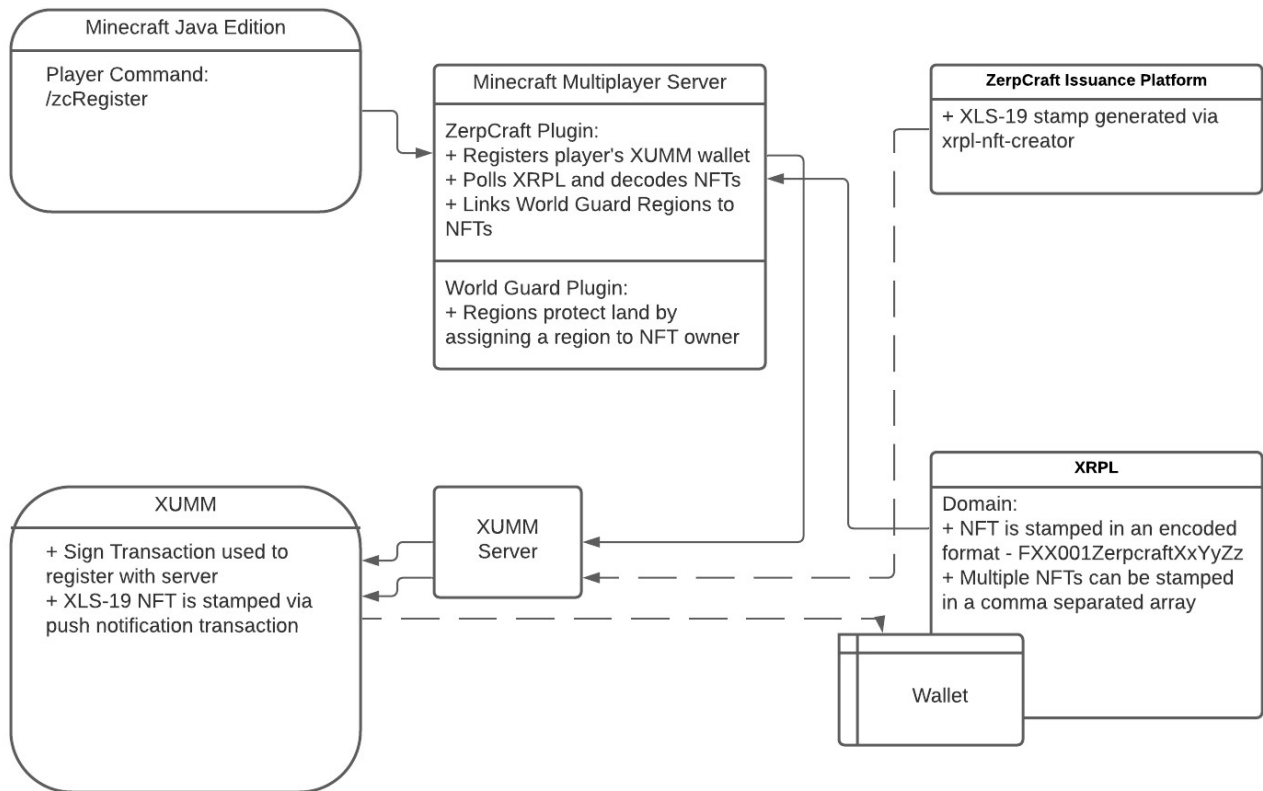
<https://github.com/XRPLF/XRPL-Standards/discussions/46>

**Pros:**

- DEX support. Buy and sell ZerpCraft NFTs easily.
- Native implementation of multiple NFTs per wallet via the NFTTokenPage object.
- NFT royalties enabled. Multiple monetization strategies and revenue splits become possible.

**Cons:**

- No smart contract capability. Complex use cases will need to be implemented on the upcoming Flare network or wait to utilize the Hooks amendment.
- Future data-rich features will need to utilize some kind of pointer to file storage such as a torrent magnet or an IPFS link.



### 3. Roadmap

This section will provide a high overview of the launch plan, transition to XLS-20d, and publishing the plugin as open source. Specifics such as dates, pricing, and participation details will be provided in a separate launch plan document.

1. The launch of the ZerpCraft community server will be accompanied by a public auction for the first ZerpCraft NFT.
2. Immediately following the auction, the Founders Sale will go live. 100 'Founders' NFTs will be minted and sold for a flat fee.
3. After the Founders Sale is concluded, Open Sale will start. This is the main land sale and is intended to be the least expensive way to buy a ZerpCraft NFT at launch.
4. Post-launch support includes stamping the XLS-19 NFTs to purchaser's wallets, any bug fixing, and the completion of commissioned builds in-game.
5. Dev work for XLS-20d implementation is completed. Upon the acceptance of the XLS-20d amendment, there will be a public Beta transitioning over to the new format. Players will need to swap their existing XLS-19 NFTs to the new standard.
6. Once XLS-20d Beta is concluded, the ZerpCraft plugin will be made open

---

source. Other Minecraft servers can implement ZerpCraft NFTs for their servers. ZerpCraft will continue to act as an NFT issuance platform for these servers for a small fee.

## 4. Acknowledgements

- ZerpCraft is grateful to the RippleX team for their generous business development grant awarded as a part of the XRPL Grants initiative.
- Twitter user @CaptainNathanA1 for being the first to install and independently test the original XRP Tipper plugin. Conversations we had planted the seeds for what became ZerpCraft's initial use case.
- Minecraft modding pioneer Nossr50 for open sourcing the McMMO plugin as well as his willingness to answer my questions when the plugin idea was in its early phases.
- Wietse Wind and XRPL Labs for the open programmability of the XUMM wallet. ZerpCraft would not be possible without the excellent dev tooling XUMM provides.
- Prime Viridian for helping me practice and refine the ZerpCraft elevator pitch as well as initially helping test the Minecraft XRP Tipper plugin.
- Github user calvincs for developing the XLS-19 standard as well as the open-source tooling for stamping the NFT to a wallet.