

# Earth<sup>64</sup>

Interoperable Data Structure

For

Digital Ownership Management V0.1.9

---

## Preface:

Web X.0 promises user-centric data ownership, data monetization, NFT mass adoption, scaling metaverses and generally an enhanced ecosystem that by design is fairer, efficient, secure, and interoperable. Third parties in communications can either be security holes, undesired friction or both. Eliminating the **dependency** on third parties will likely result in a competitive landscape amongst the **service layers** that have ongoing incentives to evolve progressively, not regressively because users do not depend on them. Earth64 is a building block. It is not the block, the same way a screw is not a building but is used millions of times in constructing buildings.

***NB: A foundational data structure that enables the implementations of a highly secure, global and efficient Ownership management system such as Non-Fungible Tokens (NFTs) that can be fully decentralized, will be without any doubt an instrumental pillar in bringing the promises of Web X.0 to life.***

## Problem:

1. At the time of this write up and up until the end of 2022, all ownership Management Systems such as (NFTs, NTT, Coins, Tokens, Digital Assets, Banks. Ledgers, Identities, Even Self Sovereign Identity) are siloed and Dependent on third parties. In this write up, we chose NFT as a use case, however, Earth64 can be applicable to more general implementations of any kind of ownership as listed above. An identical NFT created on the Cardano blockchain could be re-created on another blockchain, say Ethereum, Algorand, or Avalanche. None

of those NFTs are globally unique, they are only unique to their respective blockchain ledger.

2. Current Ownership management systems depend on a ledger. By construct, ledgers can easily be cloned - even those that claim their blockchain doesn't fork. Once such an event occurs, any supposedly 'unique' NFT that existed on that ledger is then duplicated into two distinct NFTs residing on: [1] the original ledger and [2] the forked or cloned ledger. This would make it possible to have an unbounded number of the same NFT, any digital asset, or anything of value that is digitized. **NOTE: Anything that can have unlimited amounts of will not hold its own value**
3. The cost of creating and managing ownership on current systems is extremely costly. Even when the fees are hidden by having them subsidized as in the early days of Ethereum, or early days of opening a bank account (free for a year, Amex pays you to use their payment system, etc.). The eventuality of minting an NFT on a ledger (*by construct*) is very expensive as it requires each replicated ledger to create a new record for every single NFT minted. There are claims that some ledger-based blockchains have reduced the fees while others criticize them for also reducing the security. Either Way, the cost to have replicated records across the globe is inherently high.

### **Solution:**

We propose a data structure to map onto an already existing **globally unique set** of numbers that has **relatability** to everyone, including those who are non-technical. And that relatability is essential. It is finite and sufficiently abundant to provide a 1-to-1 relationship so that, by design, a unit can only belong to a single owner in every block of time without having to depend on any third party.

While current NFTs are hyped up, the benefits of a data structure pillar will not only solve the current hype but will also:

- Enable implementations of near zero-cost minting of NFTs at a global scale.
- Provide the highest security available to modern cryptography [ECDSA, SHA..]
- Revolutionize the future of ecommerce.

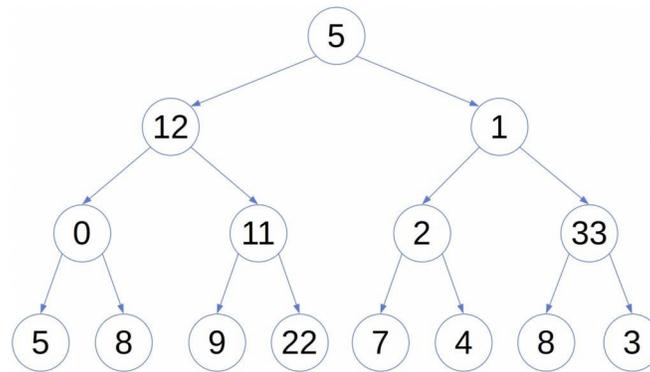
- Disrupt the entire rails of finance [DeFi, Smart Contracts].
- Increase the chances of success of Web 3.0.
- Deliver efficient self-governed systems to maximize efficiencies [DAOs].

## Why?

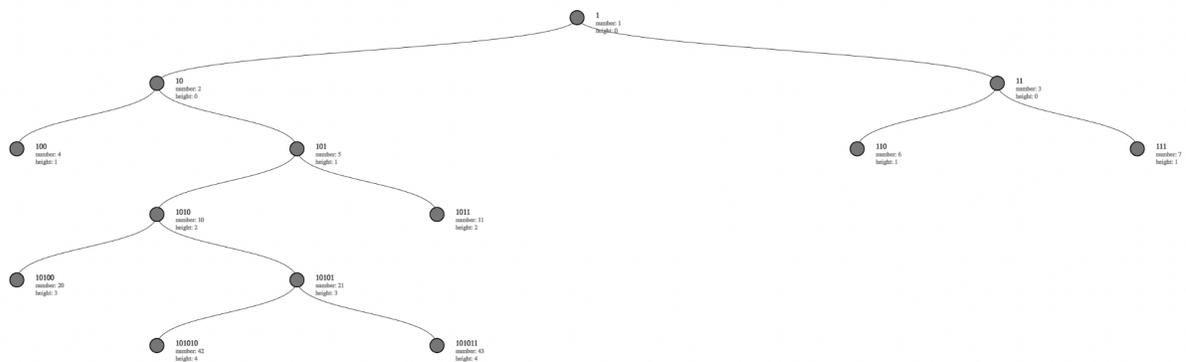
While this may seem geared to exclusively solve existing applications built on NFT management systems, that's only a use case and isn't the whole thing. The data structure is designed to reduce the friction of ownership management by several orders of magnitude in a fully decentralized setting. This data structure can help proper implementations to revolutionize the entire digital wallets ecosystem. It is simply a piece of the puzzle like any development tools. One simple application folks can think of at the current time being is how you can have fractional ownership of pennies, or micro-carbon credits, or anything of miniscule value.

## Proof:

This set of  $2^{64}$  globally unique numbers form the basis of the Earth<sup>64</sup> data structure. They are derived from two globally accepted numerical systems denoting Space and Time: [1] The geospatial coordinates of Planet Earth and [2] Coordinated Universal Time (UTC), the time standard by which the world regulates clocks and time. Anyone can easily relate to the geospatial coordinates taken to the 7th decimal place. Such coordinates point to an area of about the size of a fingertip on the surface of Planet Earth. To maintain uniqueness, the data structure is time-stamped algorithmically. This means that any other future duplication would immediately be identified as copies (*assuming time-machines are still not possible*). We subsequently took this entire set and mapped each and every fingertip-sized area to 64 bits (*ones and zeros*) which are effectively the leaves of the data structure. The data structure itself is a 64-level Binary Search Tree ("BST"), which is fully saturated, balanced, and complete.



This example above helps to visualize a fully saturated and complete BST tree, where node at the top is the root and the nodes at the bottom are the leaves. The one below, however, is a branch expansion on an actual Earth64 data structure:



These simplified snapshots of the binary tree can be expanded from the root down 63 levels to the leaves, and can be easily understood by computer scientists. In this example let's suppose you are the owner of genesis wallet 101. Anyone can immediately deduce your ability to prove that 101010 and 101011 belong to you (*notice the prefix of children always starts with 101, likewise with all grandchildren and great grandchildren, etc.*). Let's extrapolate the binary tree to cover the entire planet where each leaf is represented by a 64-bit string (64 ones and zeros) and each and every 64-bit string has an algorithmically 1-to-1, deterministic relationship with the geospatial coordinates of every ~rectangle defined by the intersection of every 2

consecutive 7th decimal longitudes and 1 consecutive 7th decimal latitude (approximately 1 square centimeter in area) representing the entire planet.

Now let's extend wallet 101 to a total of thirteen levels down and call the wallets at that level "genesis wallets". At Level 13, covering the entire surface of Planet Earth takes about 3,000 BST branches. However, it turns out only a subset of 1,701 BST branches happen to be pointing at international waters and glacial regions that are not owned by any one specific entity. Therefore, those genesis wallets will each have a finite yet abundant amount of approximately 2 quadrillion Sato-Servers. Their number, however, cannot be more than 1,701. Sometimes we refer to these wallets as bundles or branches (*like a branch of the tree because effectively they are branches of our BST*). As a result, each branch has 51 levels and contains  $2^{50}$  of those leaves.

While the name of each leaf is a unique 64-bit string, it has the ability to hold a tiny 64KB file. This file cannot be further divided nor split. We named it *Sato-Server[0]*. We also named this data structure Earth<sup>64</sup>.

Developers can insert any and all the parameters that are crucial for NFT containers into this file and we add the chain of cryptographic proofs inside this file if they ever change ownership. Those proofs are standard Merkle proofs related to those files and the protocol(s) chosen to transmit the ownership accordingly. In fact, every point in that tree can be a single *Sato-Server[Y]* that owns two others, *Sato-Server[Y']* and *Sato-Server[Y'']*, below it, except the leaves that are *Sato-Server[0]*.

By construct, each and every one of those genesis wallets can be interoperable with each other without having to rely on any third party and are hence fully independent—which introduces deterministic benefits including unprecedented efficiencies.

### **Distribution of the Bundles/Branches:**

Initially we started providing licenses to the software (the bundles of those 2<sup>51</sup> Sato-Servers) using traditional purchase order methods. During this process and prior to release, we were able to allocate 369 bundles in less than a year to over 100 customers/community members.

These presales started accruing into revenue on March 17th 2022 which is the time we automated the entire supply of software licenses into a DAO that has a smart contract which autonomously regulates its own listing price of a software license. While demand grows it increases its listing price by 1%, otherwise it does not release any licenses (i.e. it limits supply autonomously) until a successful new purchase at the preset 1% increase, and this percentage drops by half every order of magnitude in price increase. This process continues to be governed by the automated smart contracts while the treasury, which was filled up by the community members in the presale, purchases a bundle for the first 379 days from release and continues to send it to its respective owner. The funds received by the smart contract at that time will be split up between revenue to respective parties and/or license fees while the majority will lend back to the treasury as most community members bought their bundles discounted.

Effectively, we bundled quadrillions of Sato-Servers into one wallet and minted that wallet into an NFT on currently available chains {Ethereum} and {Algorand}. A maximum of one single bundle of Sato Servers represented by one NFT could be released per day at that newly increased price, and available to anyone for purchase (though it is *mainly geared towards developers*). The public key owner of the NFT is the owner of that genesis wallet. If folks choose to purchase the NFT directly from Ethereum, the price is automatically adjusted to USD via several independent oracles. Since the first 369 were sold out, the next one that will be set by the smart contract will be over \$200K USD and it won't ever sell for less by the smart contract. If it sells for more, then it mints more and can not mint more than one per 24 hour period. However, if folks opt to pre-purchase at the list price as it increases X% per day, they can do that but won't receive the license until the day the NFT is minted. This can be useful for folks to reserve their spot so they don't miss out to anyone else.

## What can be done with those NFTs?

***NB: those NFTs used to distribute bundles are not to be confused with the data structure that can enable the management of NFTs at scale; those are already existing NFTs in ledger-based blockchains that are connected to the global payment rails. The reason for this project to go to market by offering this method as an added method to provide licenses is simply that it is more efficient than the traditional Purchase Order method, etc. This doesn't mean this project doesn't provide Purchase Orders, it can, but it can also sell those licenses via NFT sales as well.***

There are only 3 things that can be done with each of those NFTs (*which represents those software licenses*):

1. Burn it to get access to the software-license;
2. Transfer it to another public key; or
3. Since it represents  $2^{51}$  software licenses, split it (at any level of the BST) into 2 in the event each half of those licenses need to be used towards a different use case.

## Product Sales:

This smart contract lists at a price X, and if sold then it waits 24 hours and lists the next NFT which represents the next software license but at a price 1% more expensive than the one just sold. When price X grows by one order of magnitude then the price increase is halved to 0.5%. This continues to operate automatically until the entire finite set of 1,701 is sold.

## Investors:

***NB: While the method described above is geared towards product distribution, using NFTs to sell the product, the owner of the entire setup can***

***be a DAO that can then be owned by many entities, which can therefore be attractive to investors and may have to comply with security laws accordingly.***

We then take this smart contract and put it under a DAO. Whoever owns the DAO owns the proceeds from the smart contract and all the inventory in it that has not yet been sold. Therefore, if someone owns 10% of the DAO, they will collect on a daily basis 10% of the revenue of newly sold inventory.

**Terms:**

<b>Earth<sup>64</sup>:</b>	The name of this project and the name of the entire Data Structure that has 2 <sup>64</sup> data points interconnected in a binary search tree that is fully saturated, complete and balanced.
<b>NFT:</b>	Non Fungible Token - In this document it refers to 2 types: <ul style="list-style-type: none"> <li>- The ERC-721 or ASA asset (depending on which blockchain it's on) that points to the software license that carries the bundle of Earth<sup>64</sup> Data Structure.</li> <li>- The ultimate NFT Management system that this Earth<sup>64</sup> software license is a pillar of and crucial in its creation.</li> </ul>
<b>Sato-Server:</b>	It's an NFT Container. Effectively, each Sato-Server can carry one single NFT in every block of time. Every data point in Earth <sup>64</sup> is referred to as a Sato-Server. If the root of the tree is represented as Sato-Sever[63, 1], the leaf level is represented as Sato-Server[0, X], where X is a 64-bit representation. For example, at the leaf level a single Sato-Server representation would look like this: Sato-Server[0, <b>1000010110010001011101010100100010101001011110101010100001101010</b> ] Similarly, the representation of a branch 13 levels from the root would look like: Sato-Server[50, <b>10100001010101</b> ]
<b>DAO:</b>	A Decentralized Autonomous Organization is a collection of software code that can have one or more Smart Contracts and it operates autonomously.
<b>Smart Contract:</b>	Software code that operates as per the rules defined at its time of release. Such rules cannot be changed. Their execution continues to be deterministic as initially set-up, ensuring to all stakeholders a deterministic outcome.
<b>Bundles/ Branches:</b>	The two terms are used interchangeably to indicate more than one Sato-Server, basically any collection of Sato-Servers can be a bundle. Most prominently, the bundles sold by the smart contracts that are also referred to as genesis wallets are at level 13 and there are 1,701 of them.
<b>Web 3.0/WebX.0:</b>	Is supposedly the next generation of the world-wide web. While there are many varying definitions of Web 3.0/Web 5.0 We call it WebX.0, the common element is that it enables users to own their own valuables without a dependency on third parties. Subsequently, this enables peer-to-peer transactions without third party dependencies.