Gild Lab.

Whitepaper 4

Using ETHGild OffchainAssetReceiptVault

GILD LAB LIMITED
Company number 13684709
Address Unit 8, 28 Mill Street, Ottery St. Mary, England, EX11 1AD

Contents

Whitepaper 4 - Using ETHGild OffchainAssetReceiptVault

GILD LAB ABOUT

We experiment with the representation of assets. The representations enable new economic structures.

Using EthGild OffchainAssetReceiptVault

OffchainAssetReceiptVault is the EthGild hybrid NFT/ERC20 solution to connect sets of custodial assets to decentralised fungible tokens.

The custodians of the assets mint and burn NFTs that represent these assets.

The assets themselves can be arbitrarily non-fungible but need to have some property that is fungible between them. It could be anything from weight, pure chemical/elemental content, revenue, etc. but there must be ONE common measure. This common measurement is used to mint ERC20 fungible tokens in the same transaction as the NFT in the same ERC1155 amount of the NFT.

Importantly the fungible property must have a liquid off chain market. Precious metals are a great example, they can already be valued, bought and sold by their pure weight in established and deep global markets. Numismatic coins on the other hand may have a pure gold/silver weight equivalent price but typically their offchain market price is determined by their collectible value, and a specific coin is not readily bought/sold for a clear spot price.

These requirements exist because the price of the fungible token is primarily set by the custodian's ability to arbitrage onchain and off chain simultaneously. If the offchain assets cannot be readily bought/sold for a clear price that is measured in the same way as the fungible tokens minted by the custodians, then there is no way for the onchain tokens to maintain price stability against the offchain assets. This model does NOT imply or require that end-user token holders have any rights or claim over the underlying assets, or physical delivery, etc. That "direct ownership" model would require users to be doing onchain/offchain arbitrage themselves, which presents logistical, regulatory, and liquidity problems that the custodian can avoid or manage themselves. There IS a form of physical allocation in that the NFT audits force the custodian to have any/all physical assets they mint fungible tokens against, but the allocation by the custodian doesn't imply ownership for the end-user.

Once the requirements are met for an asset the custodian can profit simply by arbitraging price divergences. There is no need for additional fees from the custodian to the end user to cover costs. For example, if the price of the offchain asset is higher than the onchain token, the custodian has sole right (enforced by the contract) to buy back the token and burn it, then sell the

assets in custody for profit. The reverse is also true, if the off chain price is cheaper than the onchain token the custodian can purchase additional stock and sell newly minted tokens for a profit until the price normalises.

The fungible tokens can be freely traded by non-custodians as any other ERC20, within the limits of an optional KYC integration. The KYC contracts use the standard interfaces from Rain protocol, a heavily audited toolkit for building economies.

The custodian of an asset can only burn its onchain NFT by burning an equivalent amount of the fungible ERC20 token in the same transaction. This ensures that in aggregate the total supply of ERC20 tokens is always exactly the weight/content/revenue/valuations/etc. across all existing NFTs.

To establish trustworthiness between the custodians and non-custodians, a set of certifiers must be appointed. The certifiers audit and certify the claims made by each minted NFT against the assets in the real world. This could be a physical audit such as commodities in a vault/warehouse, financial audit of a business, fine art appraisal or anything else. The certifier publishes their certification that expires at a specific date/time. If the system does not receive a new certification before the expiry then all onchain assets are immediately frozen until a certifier signs off on an extension. The system freeze eliminates the ability for the custodian to arbitrage or collect other fees, so they are strongly incentivised to maintain the system through audit cycles.

In the case of a system wide freeze it is likely that some limited transactions will be needed to repair the internal ledger to a state that a certifier will be willing to unfreeze. A set of handlers can be appointed to receive and send frozen funds in a controlled way. Ideally handlers are rules-based smart contracts, but can also be trusted entities controlling a wallet directly.

Roles & responsibilities

OffchainAssetReceiptVault is fundamentally a custodial contract. The NFTs it mints point to something off chain, so a huge amount of trust is placed in the custodians, certifiers and handlers to maintain the integrity of the system.

Custodial off-chain assets face a very similar problem as oracles such as Chainlink that must post data (such as a price feed) onchain and then everyone else is expected to trust that data. If the oracle could profit for themselves, or damage someone else by manipulating data, there will always be the temptation built into the system to lie.

Consider USDT (Tether) as an example of a major “just trust me” style custodial asset. Tether operated for years out of Hong Kong without any public audit and ever shifting banking partners (currently banking appears to be Bahamas based). As an ERC20-only solution, there is absolutely no record onchain of what justifies the minting and burning of tether.

OffchainAssetReceiptVault is still fully custodial like Tether and other stablecoins but sets the bar higher for transparency re: the underlying assets.

There is a non-custodial flavour of EthGild that has no admins but that works very differently and is out of scope for this document. The non-custodial EthGild contract only supports wrapping existing onchain assets into new onchain assets. Offchain backing for assets is always custodial in nature.

As the assets are off-chain and custodial, the contract is written to prioritise flexibility of administration with a fine-grained set of admin roles that can control every aspect of the contract logic. The contract can be configured to be very centralised with many admin keys held by one or few entities, or very decentralised with sufficiently independent or even codified (smart contracts) admins.

Every role is actually a pair of roles. The admin and the executor.

When a new OffchainAssetReceiptVault contract is deployed an initial admin can be set that is the admin for every role. Initially there are no executors. It is strongly recommended that the initial admin delegates each of the admin roles to more appropriate addresses and renounces its “superadmin” powers as soon as possible.

An admin of a role appoints/revokes executors and other admins, including themselves. This means that any admin can unilaterally and permanently remove all other admins, or even all admins entirely for their role. Clearly the private keys for admin roles must be handled very securely such as an M of N multisignature setup or even a formal governance contract that can only action the results of onchain voting.

The executor of a role merely executes the day to day operations associated with that role. Connectors connect, certifiers certify, etc. There can be many executors assigned to any role, as set by the role admin(s). Executors generally share responsibility over their task, a single rogue connector can poison the contract with forgeries, a single rogue certifier can freeze the system, etc. As it is entirely possible to achieve M of N control over private keys, this is expected to be achieved outside the logic of the contract. A single executor key may represent a group of many off-chain entities. An executor may also be a formal governance contract enforcing arbitrary workflows for actions.

If M of N schemes are insufficient for a sound trust model, consider deploying several independent contracts with meta-fungible tokens and focussing on liquidity rather than trust.

For example, consider several competing gold vaults who are all issuing gold tokens and all believe their own vault is the most secure and best audited; that everyone else is inferior. If every vault is assigned to the minter role on a single contract then they must all trust that none of their competitors will ever lie or lose assets. This clearly cannot scale up to many vaults across many jurisdictions globally as that level of trust simply doesn’t (and shouldn’t) exist. Instead, each vault

can use the contract factory to create a unique CertifiedAssetConnect contract for themselves, with their own appointed admins. Then entities who are willing to trust at least 2 of the vaults at once can be an LP (Liquidity Provider) between pairs of ERC20 tokens. This is exactly how stablecoins such as UDSC, USDT, etc. already work on fixed-sum DEXes (decentralised exchanges) like Curve protocol. If some vault is compromised then only the holders and LPs of their token are hurt, other vault tokens safely continue to operate independently.

All the role admins and executors, as well as the configuration that they set is all public information onchain. This allows token holders to always review for themselves how centralised or decentralised a given OffchainAssetReceiptVault contract is.

Connector

Connectors “connect” off-chain assets to onchain tokens. I.e. They mint matching 1155 NFTs and ERC20 tokens.

There is a single connect function on the contract that takes an amount of 1155/20 to mint and some data that feeds into audit reports for certifiers to review. Most likely the data would be something like an IPFS hash that can be fetched and processed by scripts rather than literal audit data.

For example, say a connector was minting a barrel of whiskey in a warehouse. The data about the barrel could contain all kinds of information relevant to an audit, but only the LPA (litres of pure alcohol) amount would be used to mint the 1155/20 tokens.

Typically a connector would also be a disconnector but it is not required.

Disconnector

The opposite of a connector. They “disconnect” off-chain assets by burning onchain assets.

The disconnector must hold both the NFT and enough ERC20 tokens to cover the full amount associated with the NFT in order to complete the burn. This guarantees that the aggregate amounts across all NFTs and issued ERC20 tokens are 1:1 at all times.

Disconnecting an asset is permanent/irreversible but a connector can always reconnect a previously disconnected asset with a new amount and associated audit data. In this way connected assets cannot be changed but they can be burned and re-minted with updated information prior to audit.

Partial disconnections are disallowed so it is recommended to connect assets in the size and configurations that they are typically acquired and disposed of. For example, a real estate fund would be better served connecting/disconnecting houses than entire suburbs.

Typically a disconnector would also be a connector but it is not required.

Certifier

The certifier can call the certify function to either extend or override the current certification period. By default a new certification will only maintain or increase the current certification expiry date, but a certifier may pass an additional parameter to force a specific (potentially shorter or even past) date.

Certifiers are expected to pass in additional data to reference an audit report that justifies their certification.

An audit process would typically look like:

- The certifier runs a script to extract auditable information from the blockchain logs for all connects and disconnects
- The certifier performs a real world audit of the offchain assets against the extract
- The result of the audit is posted onchain with the certify function

Multiple parallel audits are supported as each certifier can submit their own reports without interfering with each other's logs (with the exception of the forced expiry override described above).

If the certification ever expires then all assets are immediately frozen for all participants except handlers. This includes a freeze on connecting and disconnecting and for all admins!

Typically a certifier is only a certifier. They should be an "arms length" participant that can be trusted to be impartial. An impartial entity probably should not hold assets or other privileged roles as this could be seen as a conflict of interest.

Handler

Handlers are the only accounts that can send and receive either the NFT or ERC20 tokens in the event of a system freeze due to lapse in certification.

A handler need not be appointed immediately or ideally ever. In the case that the audit process breaks down and the certification cannot be granted the certifier should provide a remediation plan. The remediation plan should list a minimal set of actions that a handler can take to restore the system to a certifiable state as quickly and directly as possible.

The handler can be either a nominated EOA (externally owned account) or a smart contract. The latter could be more trustworthy as it is purely rules based, but the former may be more practical in a time sensitive situation.

It may be required that the handler is also appointed as a connector and/or disconnector to repair discrepancies in the onchain and off-chain overall asset supply.

Once the system is repaired and certified the handler role should be revoked/renounced.

Tierer

Tierers can define (or remove) standard Rain protocol ITier restrictions on transfers for both the NFT and associated ERC20.

ITier is an interface that allows for up to 8 membership levels (e.g. bronze, silver, gold, etc.) to be assigned efficiently to any address.

One common use case for custodial assets is to handle regulatory requirements such as KYC/AML restrictions. Rain protocol provides a standard Verify contract that is backend-agnostic to allow KYC/AML approvals to feed into ITier contracts.

ITier can also function as a “block list” rather than an “approve list” for more decentralised asset management. For example, USDC adopts this model, allowing all addresses to transfer by default and freezing only accounts explicitly flagged by law enforcement.

If a user already holds tokens and is removed from the requisite tier, either because they lost the tier or the tier contract itself changed, then their assets are frozen but remain on their address. There are two ways the assets can move from this point. Either the user is reinstated to the minimum tier or a confiscator takes the frozen assets (see below).

Tierers do not themselves define the users who can interact with the tokens, rather they define the contracts that do implement access restrictions, and the minimum tier that must be held for access.

A highly centralised system can tightly control and even change their tiering logic over time. A more decentralised system can set up tiering and then renounce the admins, or even completely remove tier restrictions.

Confiscator

As all offchain assets have some custodian there will be some regulatory environment that the custodian operates within. Typically this implies the possibility of legal actions such as sanctions being placed on token holders. Under extreme circumstances it may not be enough to simply freeze assets by removing the holder from the relevant tier contract. In some cases the assets may

need to be confiscated from the token holder then somehow processed (e.g. burned, set aside or redistributed) by the custodian.

There is a confiscator role that can forcibly take frozen tokens, both the ERC20 and ERC1155. This is obviously a highly sensitive role and action so confiscators cannot forcibly take tokens from unfrozen assets. This provides some protection against some malicious actor compromising the confiscator and arbitrarily stealing assets from users. First some attacker must compromise the tier handling before they could manipulate the confiscation process. Ideally the real world entities managing the confiscation and tiering are independent and arms length for maximum security. If there is no tier contract set then the confiscator is free to confiscate any asset from any address, so this is potentially a less secure configuration.

Confiscation is a simple transfer, the confiscator calls the relevant function on the smart contract and the assets are transferred to themselves. Confiscation bypasses normal transfer access requirements so confiscations are still possible during a failed audit. The best defence against a rogue confiscator is a well maintained tier contract.

Scenarios

The high level description of tokens and roles only doesn't make it immediately obvious how various likely, or even unlikely scenarios can be handled. A discussion follows of various situations and how the various roles can address each.

Cryptographic key loss

Let's get this one out of the way first as it is probably the most obvious thing that can go wrong.

As the OffchainAssetReceiptVault contract is controlled by admin keys it is relatively easy (compared to completely decentralised contracts) to end up in a bad state.

Consider for example a simple catastrophic loss of all certifier and certifier admin private keys. At some point the contract will freeze and nobody can unfreeze it.

Let's assume the worst case scenario, all the admin keys have been lost.

In this case we have no choice but to deploy a new OffchainAssetReceiptVault contract with a new set of admin keys and perform a migration.

The migration can be facilitated by:

- Connect new assets that are duplicates of the connected assets in the damaged contract

- Taking the newly minted ERC20 tokens and depositing them in a contract that will trade the old ERC20 contract 1:1 for the new tokens

- This exchange contract will have no way to extract the old tokens, they are effectively burned

- This exchange contract may not be able to receive the old tokens, in this case it may rely on a more manual snapshot style approach to allow users to claim the migrated token from their old balances

The two major risks for end-users in this scenario:

- Phishing attacks where an attacker tries to take advantage by creating a fake new contract

- The phisher cannot gain any new tokens for themselves, but they can convince users to burn old tokens for no new tokens

- Depending on circumstance users may be able to prove their innocence and have the new token manually sent to them 1:1 for phished balances (e.g. signing a message with the private key that sent the phished transaction)

- Phishing attacks have well established mitigations (clear messaging, trusted domains, etc.)

- Tokens are somehow unable to be transferred from the old contract for certain users

- The custodian should not assume that 100% of tokens deposited in the migration exchange contract will be claimed or are recoverable directly by end users

- There should be some manual migration path, perhaps activated only after 6-12+ months, to allow for unforeseen edge cases

- If the custodian somehow decides old tokens are truly unrecoverable (e.g. sent to some address like 0) then they could choose to distribute/sell new tokens, but this would need to be done with extreme caution as it is very difficult or impossible to know if crypto funds are truly lost or simply haven't moved in a long time (e.g. there are bitcoin that haven't moved in 10 years that are still considered recoverable by the owners)

Malicious cryptographic key use

In the case of any admin or executor acting maliciously or their key being compromised to a malicious actor, the attacker can commit arbitrary fraud within the bounds of their role.

Some examples of things that can happen unchallenged within a single transaction:

- Certifier can freeze the entire system by overriding current expiry time

- Connector can mint and sell unlimited ERC20 tokens

- Tierer can swap out new allow/block lists for token transfers

- Role admins can remove all executors and admins including themselves from their role, permanently removing that functionality from the contract

This is the worst case onchain scenario as a mere migration to a new contract (see above) may not be sufficient to restore trust and will not restore stolen funds. In the case that a malicious connector infinitely mints fraudulent ERC20 tokens they can directly steal end-user funds by selling on the secondary market.

As there is no remediation for stolen funds after the fact, our efforts need to focus on paranoid prevention rather than hope for some cure.

Sadly this is not uncommon, just today the Ronin network was hacked for $600 million by an attacker gaining access to 5 of 9 validator private keys. Clearly when the stakes are so high, even a large M of N multisignature may be insufficient security.

The more value embodied by the token system, the higher we need to set the threshold for an attack. Just like physical systems, the cost to attack needs to be much higher than the reward for attacking.

Fortunately multi signature wallets are not the final word on secure contract operation. All roles on the OffchainAssetReceiptVault contract can be assigned to smart contracts for increased security over raw keypairs. For example Open Zeppelin offers a set of modular governance contracts that allow DAOs (decentralised autonomous organisations) to propose, vote and execute actions on contracts with configurable quorum, time delays, etc.

As well as increasing decentralisation a DAO with a structured proposal/execution workflow improves security over a simple M of N setup by:

- Allowing an open ended set of participants as anyone who holds the governance token can make and vote on proposals

- Enforcing time delays on any action between the proposal, voting period, and execution, so that users have the opportunity to protect themselves from actions they see as malicious or unfavourable

- Potentially decreasing correlation between signatories by making voting rights easily tradeable (e.g. public vs. private shareholder voting in a traditional context)

I'm not aware of any study that attempts to quantify it exactly, but typically for appropriately configured DAOs we see governance attacks (malicious actor attempts to acquire majority voting) succeed less often and less catastrophically than directly stolen private keys.

Unrecoverable asset losses

While nobody wants to think about it, there is always the chance of catastrophic off-chain asset losses. This is the worst case off-chain scenario. Nothing is completely immune to all natural and manmade disasters, so we need a way to wind down the system as gracefully as possible.

Ideally the assets would be insured by the custodian to cover losses but we have to consider the case that the asset value can only be partially recovered.

In this case the certification process cannot succeed as too many NFTs are dangling/inaccurate references to damaged/lost assets to be salvaged by the custodians.

The ERC20 assets will need to be bought back by the custodian for some cents on the dollar discount. This buyback price can be calculated simply as:

insurance payout / total ERC20 supply

There's no way that this can be facilitated by an algorithmic price curve based DEX so an order will need to be placed on the Rain DEX at the discounted price. The Rain DEX supports orders owned by smart contracts, so the entire insurance payout can simply be deposited with the dex against an unmodifiable buyback order, either as a lump-sum or in batches to mitigate smart contract risk. In this scenario FT holders will simply sell their tokens into the order as there is no other use for them after this point. Even if the custodians wind down their operations the onchain order will remain in perpetuity for as long as the underlying blockchain operates, so FT holders will always be safe to claim their payout.

Ideally a handler admin can establish the order book as a handler so that the final winddown buyback can persist after the system is frozen due to the uncertifiable conditions. A certifier may even want to force an early expiry on the system to bring funds back to the handler as quickly as possible.

If no handler can be appointed then the DEX buyback will only be functional until the certification expiry. In this case the DEX buyback should be listed as cancellable so that when the system freezes the custodian has the option of distributing any remaining funds pro-rata directly against frozen token balances.

Cycling assets

Custodians may not hold perfectly static eternal assets. While a gold bar in a vault or a house may be easy to consider, there are many assets such as agricultural crops that have natural life cycles.

At first glance it may seem easy for connectors and disconnectors to work together to buy back ERC20 assets from the market, burn the NFTs for the old assets and connect new assets for new

assets. The problem with this is that liquidity for any tradeable asset starts high and then quickly drops as the market runs dry, leading to huge price spikes.

Said another way, as the ERC20 holders notice what is happening, they have every incentive to ask for astronomical prices knowing that the old NFTs cannot be disconnected until the disconnector holds enough ERC20 tokens to cover the NFTs being burned.

This can be solved very simply by ensuring the connection of new assets happens first before old assets are disconnected. In this case the newly minted ERC20 tokens can simply be handed from the connector to the disconnector directly without ever touching the secondary market.

The key here is to ensure that the connector and disconnector can coordinate the cycling before the next certification audit.

The primary risk is that the minting and burning will not be complete before the next audit, forcing the system into an uncertifiable state until all the previous cycle's NFTs are disconnected.

Maintaining a price peg

There's no assumption that the fungible ERC20 tokens maintain a price peg with the off-chain markets for the assets in custody. It is definitely implied and expected due to the profitable arbitrage both above and below the peg for the custodian, but the fidelity of the peg relies on the actions of the custodian. This may be a desirable property of the system so it is achievable given some additional conditions:

- There is a liquid secondary market for the ERC20 token that allows efficient arbitrage

- The custodians can participate in profitable arbitrage through off-chain markets for the assets in custody

- The custodian actually does take an active role in buying and selling the offchain assets on the time scale that the peg is expected to be maintained over

The final point is important. It is entirely possible that a custodian has the ability to buy and sell off chain assets reliably over days/months/years but no ability to maintain a spot price over minutes/hours/days. In this case the price will fluctuate organically as traders speculate against each other between the orders placed by the custodian.

Achieving liquid markets for any asset is non-trivial in general and out of scope of this document. If an off-chain liquid market already exists then an on-chain market can be established through either a standard LP based DEX (requires correlated token pairs) or a rain DEX order book (requires a market making strategy). For example, if the custodian is willing to use or provide a price oracle they can create a rain order that automatically tracks a spread of +/- some % around the oracle price. This will buy and sell onchain around the peg without any further management

by the custodian (e.g. no need for bots etc.). If the peg breaks completely through the onchain order due to exhausting the custodian's funds in the order, the custodian can continually withdraw from the DEX, then buy/sell off chain, then deposit more funds from the off chain trade until the peg is restored (at a profit for the custodian defined by their order spread).

In the case where liquidity for trades exists for the off-chain and onchain asset then any wallet with the connector role can decrease the ERC20 price when it goes above peg by buying the off-chain asset, minting the ERC20 token and selling the token. This will be profitable as the off-peg token is worth more than the asset that mints it. Similarly any disconnector that holds NFTs can increase the ERC20 price when it goes below peg by buying the on-chain asset, burning it, then selling the disconnected asset for a profit on the off-chain markets.

Recoverable asset losses

It may be the case that some individual asset is lost or damaged somehow. In this case the NFT that points to it becomes a "dangling reference". That is to say that the next certification won't pass the audit because the certifier will have NFTs in the extract that can't match an off-chain asset.

In this case the NFT needs to be burned, either permanently or to be reconnected at a lower quality/amount to represent the damages.

This is the "teeth" of the system that forces the custodians to pay for damaged and lost assets off-chain.

If a liquid market does not already exist for the onchain token the Rain order book can be used to list a one-time buyback of ERC20 tokens to cover the burn.

Winding down the system

It is not required that the custodial assets are held "forever". It is entirely possible that the custodian may only mint and manage offchain and onchain assets for a specific period of time, a season, year, decade etc.

For example, barrels of whiskey mature over the course of 8-12 years typically. The custodian may hold the barrels in a bonded warehouse until maturation but fully intends to sell 100% of all barrels at a specified maturation date to an offchain buyer (for a profit that more than covers the storage costs over the prior decade).

This presents a problem for the fungible tokens on the secondary market. The custodian wants to distribute profits by buying tokens back at, for example, 2x their original sale cost. Secondary markets for crypto assets are typically powered by price curves rather than order book style where the buyer/seller can dictate a specific price.

The custodian needs to establish a primary market for the fungible tokens where the price per token is:

- Fixed according to the final sale price of the offchain assets, e.g. 2x mint cost

- Respected so that speculators cannot manipulate the price by pumping and dumping liquidity on an AMM style DEX

- Will be permanent and always available to FT token holders even if they want to sell into the offer several years after the offchain sale has been finalised by the custodian, without further effort/permission on behalf of the custodian

In this case the Rain DEX can be used exactly as in the case of total asset wipeout, but the funds for the order are backed by the total revenue of the offchain sale (minus the custodian's cut) rather than a fractional insurance payout. The order can be placed by a smart contract that itself has no logic or ability to remove the order, this means the end-users are fully protected once the revenue funds are deposited onchain.

The mechanics are exactly the same as the insurance recovery:

Price per token = total revenue / total tokens in circulation

The difference is that here we expect the price per token to be greater than the cost the end-users paid for the tokens when they were minted (profit!) whereas in the insurance case we expect the price per token to be less than or equal to the mint cost.

The custodian after making the offchain sale can immediately mint new tokens for a new batch of replacement barrels to mature in the same warehouse space. These new custodial tokens are a completely new contract, with a new (or identical) set of users and contracts for all the management roles, and can be put up for sale in a standard Rain sale. The end users holding the old tokens can immediately sell their old tokens into the DEX order, then roll as many or as few funds from their sale into buying new tokens from the new contract or just exit (to be replaced by new users hopefully).

New assets / Replacing underlying assets (1155s) while keeping the same 20 supply

- There is no "update" for the 1155 so it works the same as cycling as above

- First mint new assets with the correct updated information then burn old assets

- Recommended to include a "replaces" or "updates" field in the NFT metadata so that it is clear that the new asset is replacing/updating an old asset, keeping the audit trail easy to follow

Change in profitability or Selling some of the 1155s / Buying back some of the 20

- The custodian is free to buy/sell the 1155s and 20s as they want for any reason, not simply to try and arbitrage a peg

- This can feed into more interesting/situational/sophisticated financial models where the 20 or 1155s are used as part of fundraising or distributions

- ONLY the custodian can mint and burn the tokens UNLIKE the non-custodial version of ETHg

- Selling and buying back can be done either on the primary market with orderbook or on any secondary market with sufficient liquidity

Asset provider wants to piggyback on an existing system

To do

Auditor lies / cheats / steals

- The ONLY power an auditor has over the system is to set the “certified until” time

- The auditor cannot steal anything onchain as they have no access to onchain assets

- The auditor cannot cheat the system as there’s only one function that they can call, which sets the certified until time

- The auditor CAN lie

- An auditor that lies needs to be removed and replaced one way or another as they are the bastion of trust in the system

- The auditor could collude with the custodian to the token holder’s detriment.

- The pair of custodian and auditor is extremely important for the token holder as part of their own due diligence because fractional holdings represents systemic risk to the token

- For as long as the custodian can lie about holding assets that they don’t hold without an auditor stopping them, the system is undercollateralized and therefore subject to a possible bank run that cannot be recovered by selling the offchain assets

- Effectively the combination of the custodian and auditor acts as a unified oracle that asserts the ability to defend against bank runs

- The auditor does not have to be a single entity, it could be a smart contract that enforces rotating through several independent auditors, or a multisig, or a DAO, etc. so there are many ways to engineer more trust-worthy systems but we cannot engineer a more trust-less system due to the limitations of oracles

- Any token holders that suspect bad audits SHOULD dump their tokens immediately, thus triggering the bank run that will leave the custodian exposed when they cannot cover the arbitrage, whether or not users DO this is another matter

- The auditor could extort the custodian by refusing to extend the time or even suddenly freezing the system even though all offchain assets are perfect

- Worst case scenario there would ideally be a snapshotter separate from the auditor who can prepare the system for an atomic migration

- The custodian can appoint a new auditor on a new contract and setup an order book order so that all token holders can claim new tokens 1:1 with their old tokens according to the snapshot

- If there is no snapshotter role the same thing can be done manually with an airdrop style token claim

- Best case scenario a new auditor can be appointed and the old auditor removed by the auditor admin, without a full system migration

Can't find an auditor

To do

Distributing revenue based on token balances

- ERC20 snapshotting is available for orderbook based distributions

- Like for like based revenue is available based on the rain staking contract

- ERC1155 snapshots are NOT available so if that was needed some kind of staking solution would need to be developed to prevent "double spend" - tier based?