HESOYAM CANOPY

A Fee-Funded Reward Economy for an On-Chain Cultivation Game

Version 1.0
September 2026

Abstract. Reward-bearing tokens usually fail at the same point: the money paid to holders is not earned, it is printed, or it is taken from the next buyer. This paper describes a protocol in which every unit paid to a participant was first paid in by somebody purchasing something they wanted. The token has a fixed supply and no mint function. Rewards are settled in a separate asset the treasury acquired with realised revenue. Ten distinct fee lines, all of them ordinary costs of playing a cultivation game, converge on a single router that splits them by an allocation stored on chain. Half of that revenue buys rewards for holders who lock the token; just over a quarter funds a descending-price bid that buys harvested output from growers. If activity stops, rewards fall to zero rather than collapsing, because nobody's principal was ever the source. We describe the economic loops that make participation reinforce itself, the randomness construction that makes outcomes ungrindable on a chain with no verifiable-randomness oracle, the ownership layer that turns land into a rent-bearing asset without turning it into a claim on later buyers, and the invariants that are asserted continuously against live chain state rather than promised in a document.

1. Introduction

The standard design for a token that pays its holders is straightforward and almost always dishonest. A contract emits new units on a schedule and distributes them to people who have locked the same units. The yield is quoted as a percentage. The percentage is large. Nothing produces the value behind it, so the only thing sustaining the price is the arrival of new buyers, and the design fails precisely when arrivals slow.

A smaller number of designs avoid emissions and instead pay from fees. This is materially better, but it introduces a harder engineering problem, because a fee-funded system must be able to prove that what it pays out is bounded by what it took in. That proof is not a marketing claim. It is a property that has to hold at every block, under adversarial conditions, including when the people running it would prefer it did not.

HESOYAM CANOPY is a cultivation game whose economy is built to make that property checkable. A player buys a bench, rents ground to put it on, plants a strain, feeds it across three windows, responds to pests, harvests it, optionally cures it, and sells the result. Every one of those actions costs something. Those costs are the entire source of every reward the protocol pays.

The game exists because a fee-funded economy needs fees, and fees need people who want to do something. A protocol that only offers staking has one fee line and no reason for anyone to generate

volume. A protocol with a game people enjoy has ten, and the people paying them are getting something they wanted in exchange.

2. What This Is Not

It is worth being explicit about the properties that are deliberately absent, because each is a design decision rather than an omission.

There is no fixed yield. Any number displayed to a participant is trailing arithmetic over fees that have already been collected. It moves every block and it is capable of reaching zero. No rate is printed anywhere in the product.

New deposits fund nothing. There is no function in any contract that moves staked principal into the reward pool. It is not restricted or guarded; it does not exist. Staked principal is denominated in the protocol token, while rewards are denominated in a settlement asset. The two cannot be confused because they are different balances of different contracts.

There are no emissions. Supply is fixed at launch. The token contract has no mint function. This is verified from the ABI in an automated check rather than asserted in prose.

There is no referral tree. Referral compensation, where present, is one level deep and paid from the protocol's own allocation rather than from the person who was invited.

The honest floor is zero. If trading and playing stop, rewards stop. Not negative, not a cascade: zero. Principal remains withdrawable and the game continues to function. This is the property that a fee-funded design buys, and it is the reason for accepting the extra complexity.

3. The Flywheel

The system is a wheel rather than a pipe. Value entering at one point makes the next entry more likely, and each loop closes back on the one before it.

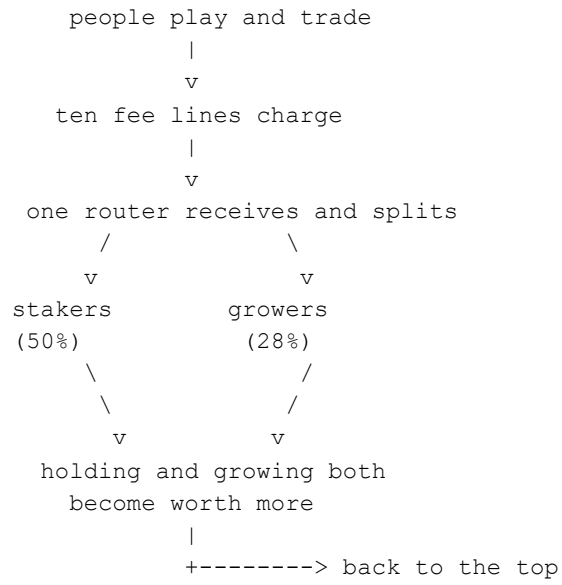

Figure 1: The primary loop. Nothing enters this diagram that was not paid in.

Four reinforcing loops run inside it.

3.1 The revenue loop

Somebody plays or trades. Fees land in the router. Half of the realised revenue buys the reward that stakers claim. Holding and locking the token therefore pays more than it did. More people hold, stake, and play, which produces more fees.

3.2 The floor loop

Fees fund the epoch budget of a descending-price buyer. That buyer opens above prevailing market price and decays across a day. A grower consequently always has somebody to sell to, which makes growing worth doing, which produces more grows and therefore more fees. The bid is not a guarantee and is explicitly not a price floor; it is a budget that runs out.

3.3 The scarcity loop

Harvested output is burned to craft cards that raise a holder's share of the reward pool. Burning removes output from circulation. Reduced supply supports the price of output, which makes growing pay better per unit, which produces more grows. This loop is what connects the two halves of the audience: a holder who never plants still wants cards, and must buy output from somebody who did.

3.4 The land loop

More grows means more benches needing ground. Ground is a fixed-supply asset whose owners collect rent. More rent makes ground worth more, which drives siting and fusing, both of which pay the router again.

4. Where Money Enters

Ten fee lines feed the router. None of them is a deposit; each is somebody paying for something they wanted.

Line	Rate	Notes
Token transfer tax	5.0%	4.0% to the creator, 1.0% to the router. Applies unless both ends are exempt.
Bench sale	100%	The entire purchase price of a bench is revenue.
Seed fee	per plant	Charged when a grow starts.
Nutrients	x3	One fee per feeding window.
Treatment	per event	Charged when responding to a pest or heat event.
Utilities	per day	Accrues across the cycle, settled at harvest.
Cure fee	per cure	Only paid if curing is chosen.
Marketplace take	4.0%	On every peer-to-peer fill.
Craft fee	per craft	Paid alongside the output that is burned.
Land rent and fusing	30% / fee	The protocol's share of rent, plus a fee to fuse two parcels.

Only the first of these depends on trading volume. The remaining nine depend on people playing, which is why the game is not decoration around a token but the substance of the revenue model.

5. The Router

A single contract receives every fee line. There is no treasury wallet in the middle and no discretion over where the money goes, because the allocation is stored on chain and must sum to exactly ten thousand basis points.

Destination	Share	Purpose
Stakers	50.00%	Buys the reward asset distributed to lockers
Growers	28.00%	Funds the descending-price bid for output
Liquidity	12.00%	Deepens the pool the protocol depends on
Operations	8.00%	Runs the infrastructure
Reserve	2.00%	Retained against failures

5.1 Conversion guards

Revenue arriving as protocol tokens must be converted before it can be distributed. That conversion is the moment at which the treasury is most exposed, so it is bounded in four independent ways.

A single conversion may not exceed a maximum size, which prevents any one transaction from dumping the entire balance. A cooldown of at least one hour separates conversions, which prevents the size limit from being defeated by repetition. Conversion is refused when the spot price deviates from the time-weighted price by more than a configured tolerance, which prevents the treasury from selling into a manipulated market. Finally, the minimum acceptable output is computed on chain from the time-weighted price and the caller may only raise it, never lower it.

That fourth guard exists because conversion is permissionless. A dead operator must not be able to halt payouts, so anybody may trigger it; but if the caller supplied the slippage tolerance, anybody could supply zero, push the spot price down within the deviation band, let the treasury sell with no floor, and buy back. The deviation check runs before the swap and constrains only the pre-trade gap, never the protocol's own price impact. Computing the floor on chain closes that gap. The amount actually received is measured by balance delta rather than taken from the adapter's own return value.

Every bound is itself bounded. The maximum conversion size has a ceiling, the cooldown has a floor, and the deviation and slippage tolerances have ceilings. This limits how far a compromised operator can loosen the protections before doing damage.

6. The Staker Rail

Half of realised revenue is handed to a vault that spreads it across all locked weight using an accumulator. Weight is the product of three terms: the amount locked, a multiplier determined by lock duration, and a bonus determined by equipped cards.

Tier	Lock	Multiplier
Seedling	1 hour	1.00x
Vegetative	30 days	1.30x
Flowering	60 days	1.80x
Canopy	90 days	2.50x

The card bonus is capped per wallet at 4,200 basis points, so no participant can exceed a 1.42x multiplier from cards regardless of how many they acquire, and at most five may be equipped at once. This exists so that the reward distribution cannot simply be purchased outright.

The shortest tier carries a one-hour minimum hold rather than none. This is not a lockup in any meaningful sense, but it closes a specific attack: with a zero-duration tier, an unlock timestamp equals the deposit timestamp, so an early-exit penalty can never apply to an exit in the same block. Because distribution is permissionless, a borrowed position could be deposited, a distribution triggered, the reward claimed, and the position withdrawn atomically, capturing the rail at no cost and no duration of risk. A one-hour minimum makes the position impossible to hold within a single transaction, so the penalty always applies.

6.1 Accumulator precision

The reward asset carries six decimals while weight carries eighteen. A naive accumulator at eighteen decimals of precision produces a quotient that rounds to zero once total weight exceeds roughly 10^{26}, at which point an entire distribution is destroyed silently and permanently. The accumulator therefore operates at thirty decimals of precision, and a distribution too small to move it at all is carried forward rather than discarded.

Carrying the exact truncation remainder on every distribution is tempting and wrong: rewards are read as the floor of weight multiplied by the accumulator, and the floor of a running sum can exceed the sum of the individual floors by one unit, which lets the final claimant withdraw a unit the vault does not hold. Discarding sub-unit dust is the correct trade, and the dust remains in the contract rather than being credited to anybody.

6.2 Exit

An emergency withdrawal path returns principal at any time, applies the same early-exit penalty, and forfeits unclaimed rewards on purpose. It has no pause, no administrative gate, and no external call other than returning the participant's own tokens. This guarantees that a participant can always recover principal regardless of what else in the system is broken or paused.

7. The Grower Rail

Just over a quarter of realised revenue funds a buyer of last resort for harvested output. The mechanism is a descending-price auction. Each epoch opens at 115% of a reference price and decays linearly to a floor of 60% across twenty-four hours. A seller may accept at any point.

Two properties matter. First, spend can never exceed the epoch budget, and the budget is only ever what the router actually delivered; the bid cannot promise money nobody earned. Second, the clock restarts only when the budget is exhausted, never on a top-up.

That second property was not obvious. Restarting the clock whenever new funding arrived meant that anybody could reset the price to the ceiling for the cost of a trivial donation, because distribution is permissionless and reads a live balance. A seller could simply wait for the price to reach the floor, poke a funding call, and sell at the ceiling instead. The auction never discovered a price and the protocol always paid the maximum. Restarting only on exhaustion means a top-up adds money without rewinding the price, and the bid rests at its floor until the budget is spent.

There is deliberately no guaranteed floor price. A guarantee would require money that had not been earned yet, which is the property this entire design exists to avoid.

8. The Cultivation Loop

Growth is a pure function of elapsed time and the care actually given. There is no ticking, no upkeep transaction, and no way for the state to drift from what the contract computes.

A cycle runs seven days. Three feeding windows open across it, each a day wide, and hitting all three is worth forty of one hundred care points. The bench tier contributes up to thirty more, representing environment control. The remaining thirty come from responding to events.

Care determines yield on a range from 55% of base at zero care to 100% at full care, a spread of roughly 1.8x. That range is deliberately wide enough to reward attention and narrow enough that sleeping is permitted. Quality is a separate scale which determines the tier of the resulting output, and an optional two-day cure adds up to twelve quality points for an additional fee.

Events are pests, mould and heat. Each fires or does not according to the grow's seed, and an untreated event that fired costs yield permanently. The response window is finite. Because the schedule is fixed at planting and visible from the moment it settles, responding is a scheduling decision rather than a coin flip.

9. Randomness Without an Oracle

This section describes the most consequential piece of engineering in the protocol, because every earlier version of it was exploitable.

9.1 The failure

Outcomes were originally derived from the participant's own committed value, the previous block hash, and the timestamp. Every one of those is known to the participant at the moment they submit the transaction. The consequence was demonstrated by two tests. In the first, a player searched their own commitment until the resulting event schedule contained no events at all, avoiding every treatment fee and opting out of the entire risk layer of the game; the search took roughly eight attempts and could be performed inside the planting transaction itself. In the second, a player waited for a favourable block and forced the maximum quality roll, at the cost of gas alone.

A commitment scheme that commits only to values the committer already knows is not a commitment scheme. It provides the appearance of fairness with none of the substance.

9.2 Why the usual answer was unavailable

The standard remedy is a verifiable randomness oracle. At the time of writing, the leading provider supports nine networks, and the deployment target is not among them; no chain of its family is. Waiting for that to change is not a plan, so the construction below was required to work without any oracle at all.

9.3 The construction

The protocol operates a hash chain. An operator selects a secret offline and hashes it many times, publishing only the final hash. Each round, the operator reveals the preimage one step back down the chain. Anybody may verify a revealed value by hashing it forward and comparing it to the previously published value.

```
secret --H--> ... --H--> h(2) --H--> h(1) --H--> HEAD
                                                   ^
                                              published once

round 1 reveals h(1);  anyone checks H(h(1)) == HEAD
round 2 reveals h(2);  anyone checks H(h(2)) == h(1)
```

Figure 2: The chain is walked backwards, verified forwards.

Neither party can steer an outcome. The operator cannot, because the chain was fixed before any participant committed, and for a given published value exactly one preimage is accepted; producing a different one requires breaking the hash function. A participant cannot, because they bind to a round that has not yet been revealed, so at the moment they decide to act there is nothing in existence to search against.

Publication of the chain head is a one-time operation and cannot be repeated. If an operator could replace it mid-flight, they could select a new chain after observing what participants had committed to, which reinstates the original attack in a more powerful form.

The round seed additionally mixes in the hash of the block carrying the reveal. This does nothing against a grinding operator, who is already bound by the chain, but it means that possession of the secret alone does not permit prediction: at the moment a participant commits, the hash of the block that will eventually carry the reveal does not yet exist.

9.4 The cost

The construction trades a liveness assumption for the grinding resistance. If the operator stops revealing, rounds stop advancing and anything bound to a future round cannot resolve. The system fails safe rather than fails open: an unresolved grow causes the harvest to be refused rather than resolved against an absent seed, because treating an absent seed as valid would produce exactly the outcome an attacker wants, namely no events fired and full marks for handling them. An abandonment path always exists so that no asset is ever permanently trapped.

This makes the revealer a required service rather than a background convenience. It should be operated with monitoring, its signing key should hold nothing else, and the secret should never reside on the machine used for deployment.

10. Land

Ground is a fixed-supply collection of 256 parcels across eight districts. It is the ownership layer of the protocol, and it is designed so that owning it is a claim on activity rather than a claim on later buyers.

10.1 Rent

A bench must be sited on a parcel to be used. Siting is a service the grower purchases for a duration at a price the parcel owner sets, and the payment splits: seventy percent to the owner, thirty percent to the router. A parcel on which nobody wants to grow earns exactly zero, forever. That is the honest floor, and it is the same property as the staking rail: payment is compensation for a service rendered, never a share of somebody else's entry.

Rent is credited to the owner's address at the moment of siting rather than accrued against the parcel. Accruing against the parcel would mean that selling mid-lease hands the buyer rent the seller had already earned.

The siting call carries a maximum total that the tenant specifies. Without it, an owner could observe a pending transaction, raise the daily rate to its maximum, and charge the tenant a multiple of the quoted price against a standing approval. The transaction now reverts rather than silently overcharging.

A caller must own the bench they are siting. Without that requirement, an attacker could bind a stranger's bench to a parcel they themselves own, locking it out of every other parcel for the maximum lease duration while recovering seventy percent of the rent from themselves, so that the griefing cost only the protocol's cut.

10.2 Supply

Two paths create a parcel and both count against the same cap: an owner-only allocation path, and a public paid mint whose entire price is protocol revenue. The public mint is limited per wallet, not as a fairness gesture but because a single buyer taking the entire supply leaves no rental market to speak of.

The only operation that changes supply afterwards reduces it. Two parcels of the same district and the same tier may be fused into a single parcel of the next tier, burning both and paying a fee. Requiring equal tiers is what makes the top tier cost eight parcels rather than four; permitting a high-tier parcel to be paired with a disposable low-tier one halves the cost of the scarcest asset in the collection.

10.3 Artwork

Parcel imagery is generated entirely on chain as a deterministic function of the parcel's district, tier and seed. There is no pinned file and no metadata server, so the artwork renders identically forever and cannot degrade when somebody stops paying a hosting bill. Rows are run-length encoded before emission, which keeps the returned document small enough to be comfortable in a wallet, and the rendering function is read-only so its cost falls on the node answering the query rather than on any participant.

11. Crafting

Cards are created by burning harvested output and paying a fee. This is the only mechanism that removes output from circulation and the only bridge between cultivating and holding.

Rarity is drawn against a published odds table that improves with the quality tier of the output burned. The draw is settled against a round that had not been revealed when the request was made, on the same principle as the cultivation schedule.

Because the output and the fee are consumed at request time while the rarity settles afterwards, an unsettled request represents value already spent. The interface surfaces pending requests prominently for this reason.

12. The Secondary Market

Harvested output is a semi-fungible asset identified by strain and quality tier. A grower may sell it into the protocol's descending bid, or list it directly for other participants at a price of their choosing. The peer-to-peer venue takes four percent of every fill, which is the ninth fee line.

The two venues serve different purposes and the difference matters. The protocol bid exists so that a grower is never stranded with unsellable output, which is what makes cultivation worth beginning; it is funded, bounded, and can be exhausted. The peer market exists so that price is discovered by people who actually want the goods, which is what makes the protocol bid's reference price meaningful in the first place. A protocol that only offered its own bid would be setting prices rather than discovering them.

Listings are settled against balances rather than promises. A fill transfers the asset and the payment in the same transaction, so a listing cannot be filled against inventory the seller no longer holds. The remaining quantity on a listing is derived from the fills recorded against it rather than accumulated by decrementing a counter, which means a replay of the same event leaves the same number rather than driving it negative.

13. The World

The game is played in a rendered environment rather than through a form. Participants walk a shared cultivation floor in first person, approach a bench, and act on it there. Every object in that environment is drawn from chain state: a bench exists because the chain says somebody owns it, and a plant is the size it is because growth is the same pure function of elapsed time that the contract computes. The renderer is a view of the ledger and never a second source of truth.

Other participants are visible and move in real time. This is the one part of the system that is not on chain, and it is deliberately the one part that does not matter. The service carrying it holds positions and nothing else: no key, no signature, no authority over any balance. Shut it down and the world still renders every parcel and every plant correctly, and every transaction still functions; the floor simply becomes empty of other people. That separation is why a compromise of the most exposed, most stateful, least audited component in the system cannot reach anybody's assets.

Acting on an object in the world opens exactly the same transaction the tabular interface would. There is no second implementation of any rule, no server that decides an outcome, and no state that exists

only in the renderer. This is a deliberate constraint on how much the game may be, and it is the constraint that keeps the security surface honest.

14. The Token

Supply is fixed at one billion units and no mint function exists in the contract. This is verified programmatically from the compiled interface rather than asserted; a check that attempted the call and caught a revert would also pass against a function that merely reverted, which is a weaker property.

A transfer tax of five percent applies unless either end of the transfer is exempt. Four percent is directed to the creator and one percent to the router. Protocol contracts are exempt so that internal movements do not erode balances.

Enabling the tax is a one-way operation. Once active it can never be disabled or altered by anybody, including the deploying party. This is deliberate: a tax that can be changed after launch is a tax that can be raised after launch.

The tax debits the sender the full transfer amount and reduces only the recipient's credit. This means no contract can be rendered insolvent by an outbound transfer, which matters because several contracts hold balances on behalf of participants.

Contracts that receive fees are written to measure what actually arrived rather than assuming the quoted amount was delivered. A contract that assumes otherwise while holding a balance on behalf of others becomes insolvent by exactly the tax on every receipt, and the shortfall surfaces only later, as a failed withdrawal by whoever happens to be last.

15. Invariants

The following properties are asserted by the test suite and, for the subset marked, re-asserted against live chain state on every continuous integration run and by an operator-runnable script.

- Claims can never exceed what was notified. Rewards are only ever credited from funds that were actually transferred in. *Checked live.*
- The reward vault always holds at least what it still owes, so the last claimant is paid the same as the first. *Checked live.*
- No mint path exists on the token. *Checked live, from the interface.*
- The allocation sums to exactly one hundred percent. *Checked live.*
- The grower rail never spends more than it was funded. *Checked live.*
- Staked principal and rewards are different assets, so they cannot be confused. *Checked live.*
- Both rails are funded from the same realised revenue. *Checked live.*
- Emergency withdrawal has no pause and no administrative gate.
- Card-derived weight is capped per wallet.
- Parcel supply is monotonically non-increasing once the cap is reached.

A separate mirror of chain events is reconciled against derived state by eight consistency checks. Three earlier checks were removed because they were structurally incapable of failing: one re-verified an arithmetic operation the writer had just performed, one restated a database primary key, and one asserted a condition every writer enforced in the same statement. A green indicator that cannot turn red is worse than no indicator, because it is trusted.

16. Architecture and Trust

The system separates into three planes with different trust properties, and the separation is the reason a compromise in one does not become a loss in another.

Plane	Holds	If compromised
Value	Contracts on chain	Loss is possible. This is why it is the smallest surface and the most heavily tested.
Mirror	Indexer and database	Displayed history becomes wrong. No value moves.
Presence	Multiplayer positions	Players appear in wrong places. Nothing else.

The presence service holds no key, signs nothing, and has no authority over any value in the game. If it is shut down entirely, the world still renders every parcel and every plant correctly from chain state, and every transaction continues to function. This is why it can be a plain socket server rather than an authoritative game server, and it is why a compromise of it cannot reach anybody's assets.

The revealer holds a key that can do exactly one thing: advance the beacon. It cannot move funds, alter parameters, or affect any balance.

Row-level security on the mirror database is used as a write boundary and never as a value boundary. No security property of the protocol depends on a database rule.

17. Security Posture

Two adversarial reviews were conducted before launch, one over the contracts and one over the application, indexer, authentication and continuous integration. Both were instructed to prefer silence over manufactured findings and to report what was solid alongside what was not.

Eleven exploitable issues were found and fixed. Each has a regression test that was originally written to assert the broken behaviour and now asserts the correction, so a reintroduction turns the suite red rather than passing quietly. The most severe were: destruction of entire reward distributions through precision loss; a zero-cost capture of the staking rail using borrowed liquidity; a resettable auction that never discovered a price; a front-runnable rent quote; unbounded administrative parameters; and, in the application, an injection sink that would have permitted a hostile token to exfiltrate the secrets participants store locally to claim their own harvests.

Neither review substitutes for an independent audit by a third party, and this document should not be read as claiming otherwise.

17.1 Residual risks

Operator liveness. The randomness construction requires a revealer to be running. Its failure pauses resolution rather than corrupting it, but a pause is still a failure.

Privileged functions. Administrative parameters are bounded, which limits how much damage a compromised operator can do, but bounds are not delays. Ownership should be held by a timelock or multisig, and this is the single largest available reduction in risk.

Sequencer trust. The deployment target has a centralised sequencer. An adversary who both possessed the beacon secret and could influence block production would have more leverage than either capability alone confers.

Economic parameters. Every rate in the design is an estimate until real participants push volume through it. They should be recalibrated from measured behaviour rather than defended because they were chosen first.

18. Fairness

In aggregate, participants pay more into this game than the game pays out. It is a rake. Stating that plainly is preferable to letting somebody discover it later.

Advantage comes from three sources: being better than average at cultivation, which is worth roughly 1.8x on yield; owning scarce assets that others pay to use; and locking what is earned. None of these is a promise of profit, and no combination of them makes the aggregate positive.

19. Conclusion

We have described an economy in which every unit paid to a participant was first paid in by somebody buying something they wanted, and in which that property is enforced by construction rather than by policy. The token cannot be minted. Rewards are a different asset from principal. The allocation is fixed on chain and sums to exactly one hundred percent. Conversion is bounded in four independent ways, each of those bounds is itself bounded, and the floor price is computed by the contract rather than supplied by whoever calls it.

The randomness that determines outcomes is drawn from a chain fixed before anyone committed and revealed only afterwards, so neither the participants nor the operator can steer it, on a network where no verifiable-randomness oracle exists. The ownership layer pays its holders from rent for a service rendered, and pays nothing at all when no service is rendered.

The design accepts a real cost for these properties. Rewards are variable and can be zero. A required service must be operated and monitored. Nothing here is a yield, a rate, or a guarantee. What it is

instead is a system whose central claim can be checked at any block by anybody, which we consider the more valuable property.