WHITEPAPER / ENGLISH

Resatoshi

Expiring Outputs and Recycling Existing Units into Future Mining Rewards

Version 1.0 · November 17, 2026 · Currency unit: RST

Abstract

Resatoshi is an independent SHA-256d proof-of-work blockchain based on Bitcoin Core. Its mainnet implements a finite lifetime for unspent transaction outputs and a recycling pool that returns expired amounts through future mining rewards.

An output expires after 5,256,000 blocks, approximately 100 years at the ten-minute target interval. Owners can renew their holdings by spending outputs before expiry and creating new outputs under their control. Outputs still unspent after their expiry block's transactions are removed from the live UTXO set, and their amounts enter the recycling pool.

Miners may claim up to 1 RST per block from the available pool, in addition to the ordinary subsidy and transaction fees. Recycling transfers existing units rather than extending the issuance schedule.

The first possible expiry occurs at block 5,256,001, approximately in 2126. Before that height, the pool remains empty and recycling rewards are zero. Recycling is therefore a long-term mechanism rather than a source of additional rewards during the network's first century.

Resatoshi combines these rules with genesis-anchored ASERT difficulty adjustment and an open maintenance approach using Bitcoin Core upstream as a common reference.

---

1. Motivation and Design Position

1.1 Inaccessible holdings

A blockchain can verify whether a transaction satisfies an output's spending conditions. It generally cannot determine whether an unused output represents deliberate saving, a lost key, failed inheritance, or abandonment.

Resatoshi does not attempt to identify lost holdings. Instead, it gives outputs a public lifetime and applies that condition consistently. An owner who wishes to retain control must spend and renew the output before expiry.

This makes permanent inactivity a defined protocol event rather than an indefinite state.

1.2 Long-term mining compensation

As ordinary block subsidies decline, transaction fees become increasingly important to mining revenue.

Tail emission addresses this problem through continuing issuance. Demurrage applies holding costs through balance decay or other charges, with details varying by design.

Resatoshi takes a different approach:

> It retains a finite issuance schedule while using expired existing units as an additional source of mining rewards.

Approach	Source of resources	Principal monetary effect
Tail emission	Continued issuance	Supply continues increasing
Demurrage	Holding charges or balance reduction	Holdings incur a recurring cost
Resatoshi	Expired outputs	Holders must renew outputs before expiry

Resatoshi does not continuously reduce an output's nominal amount. The output remains intact until spent or expired, although renewal may require a transaction fee.

At the target interval, the maximum recycling payout is 52,560 RST per 365-day year—approximately 0.25% of the 21 million RST supply ceiling. This is a payout-rate limit on existing units, not an annual issuance rate.

1.3 State and technical longevity

Outputs that are never spent can remain part of a blockchain's current state indefinitely. Expiry places a finite lifetime on these old entries.

The same mechanism can provide an end date for live outputs using a retired cryptographic mechanism, once creation of new outputs of that type has stopped.

Long-term operation also requires maintainable software. Resatoshi uses a shared upstream reference and public source history so that maintenance can continue beyond the original contributors.

---

2. Mainnet Parameters

Our mainnet uses the following rules.

Parameter	Value
Consensus	Proof of work
Mining algorithm	SHA-256d
Target block interval	600 seconds
Difficulty adjustment	Genesis-anchored ASERT
ASERT half-life	172,800 seconds: two days
Initial block subsidy	50 RST
Subsidy halving interval	210,000 blocks
Issuance	Bitcoin-style halving schedule with integer truncation
Supply ceiling	21 million RST; scheduled issuance is slightly below this ceiling
Smallest unit	0.00000001 RST
Output lifetime	5,256,000 blocks
Maximum recycling reward	1 RST per block
First possible expiry	Block 5,256,001
Genesis header timestamp	September 4, 2026, 00:00:00 UTC

Genesis block hash:

```
0000000379813245fcdd4600969c6f74684f14433badf837be5b8ffa560174ec
```

The genesis header timestamp identifies the chain’s reference time; it does not establish the exact moment public mining began.

Sources: chain parameters, consensus parameters, and subsidy calculation.

---

3. Output Lifetime and Renewal

3.1 Scope

Expiry applies to individual unspent transaction outputs, or UTXOs, rather than entire wallets or addresses.

Ordinary transaction outputs and coinbase outputs are subject to expiry once present in the live UTXO set. Provably unspendable outputs, such as scripts identified as unspendable through OP_RETURN, do not enter that set and do not later fund recycling through expiry.

A wallet may contain outputs with different creation heights and expiry dates. One transaction does not renew every output in the wallet: the outputs being renewed must actually be consumed.

3.2 Exact boundary

Let an output's creation height be h, and let:

```
L = 5,256,000
```

The output can still be spent in block:

```
h + L
```

Transactions execute first. If the output remains unspent afterward, it expires.

An output created at height 1 therefore has its final spending opportunity in block 5,256,001.

3.3 Renewal

An owner renews an output by spending it and creating a new output under their control. Sending funds back to a self-controlled address is sufficient.

The new output receives a new creation height and lifetime.

Renewal requires block inclusion. Broadcasting a transaction before the deadline does not itself prevent expiry.

3.4 Why approximately 100 years?

The lifetime allows long-term saving and inheritance while placing a finite horizon on unattended outputs.

A shorter lifetime would recycle inactive amounts earlier but increase renewal frequency. A longer lifetime would reduce renewal frequency while delaying state retirement and recycling.

One hundred years is a design choice balancing these concerns, rather than a mathematically proven optimum.

3.5 ASERT and the calendar

At the target interval:

```
5,256,000 × 600 = 3,153,600,000 seconds
```

This equals 100 years using 365-day years.

Our ASERT implementation adjusts difficulty against a genesis-anchored schedule. When block timestamps place the chain ahead of or behind that schedule, difficulty responds in the direction that counteracts the deviation.

The two-day half-life controls this response; it is not a two-day retarget window.

This provides corrective pressure against persistent schedule drift. Expiry nevertheless remains height-based, so its precise calendar date depends on actual block production and timestamp behavior.

Source: difficulty adjustment.

---

4. Recycling Pool and Reward Accounting

4.1 State transition

The recycling pool is explicit chainstate, not an administrator-controlled wallet.

For a block, define:

- P: pool balance before the block;
- E: amount expiring after that block's transactions;
- B: ordinary subsidy plus transaction fees;
- V: total coinbase output amount.

The claimed recycling reward is:

```
C = max(0, V − B)
```

A valid block must satisfy:

```
C ≤ min(1 RST, P + E)
```

The resulting pool balance is:

```
P′ = P + E − C
```

4.2 Same-block availability

Amounts expiring in a block are available to that block's coinbase reward.

A miner can therefore claim recycling rewards with a previously empty pool if enough outputs expire in the same block.

4.3 Optional claims

Miners may claim less than the available recycling reward, including zero. Unclaimed recycling amounts remain in the pool.

Underclaiming the ordinary subsidy or fees does not credit those unclaimed amounts to the pool. Only the coinbase amount above the subsidy-plus-fee allowance is treated as a recycling withdrawal.

4.4 Distribution cap

The 1 RST limit prevents one block from collecting an entire large expiry event.

A pool of 1,000 RST supports 1,000 blocks of maximum claims if there are no additional inflows.

The cap distributes rewards across blocks, not equally across miners. A miner producing multiple blocks can collect multiple rewards.

4.5 Reorganization recovery

Block undo records preserve the preceding pool balance and outputs removed through expiry.

Disconnecting a block restores those expired outputs and the previous pool balance alongside ordinary transaction rollback.

Source: recycling state transition and accounting.

---

5. Activation Timeline and Economic Scale

5.1 The first century

Recycling is enabled in the protocol from the start, but no output can expire before completing its lifetime.

Stage	Height or approximate time	Recycling status
Chain origin	Genesis, September 2026 reference time	Pool starts empty
Before first possible expiry	Through height 5,256,000	Pool empty; recycling reward zero

Stage	Height or approximate time	Recycling status
First possible expiry	Height 5,256,001; approximately 2126	Surviving outputs from height 1 can expire
Later operation	Subsequent blocks	Further expiries can fund capped rewards

The first possible expiry does not necessarily produce a positive inflow: outputs originally created at height 1 may already have been spent.

5.2 Subsidy at first expiry

At height 5,256,001, the ordinary subsidy has undergone 25 halvings.

With integer truncation to the smallest unit, it is:

```
0.00000149 RST
```

If the pool is funded, a recycling reward of up to 1 RST can therefore account for nearly all non-fee block compensation at that stage.

This is the intended long-term role of recycling. During the preceding century, block compensation comes from ordinary issuance and fees.

5.3 Pool-duration scenarios

At ten-minute blocks, a 365-day year contains 52,560 blocks. A continuously funded pool can pay at most:

```
52,560 RST/year
```

With no new inflows and a full 1 RST claim in every block:

```
Pool duration in years = Starting pool in RST / 52,560
```

Using 21 million RST as a rounded supply reference:

Hypothetical amount already in the pool	Pool balance	Duration at maximum payout
1% of reference supply	210,000 RST	About 4.0 years
5%	1,050,000 RST	About 20.0 years
10%	2,100,000 RST	About 40.0 years

These scenarios do not estimate lost holdings or assume simultaneous real-world expiry. Actual inflows depend on creation heights, spending, and renewal.

5.4 Funding conditions

If owners renew nearly all outputs, little enters the pool. If withdrawals exceed inflows, the pool declines and may empty.

Recycling does not create a permanent entitlement to additional rewards. Its availability depends on expired existing units.

---

6. Supply Conservation and Intended Benefits

6.1 Conservation

Expiry transfers an amount from live outputs to the pool. Recycling payouts transfer it from the pool to coinbase outputs.

For these transfers alone:

```
Live UTXO amount + pool balance
```

is conserved.

Ordinary issuance, fees, unclaimed allowances, and unspendable outputs require their own accounting. Recycling must never leave the same amount represented in both a valid output and the pool.

6.2 Return of inaccessible amounts

Amounts locked behind lost keys can eventually return to circulation through mining rewards.

This increases accessible holdings without extending the issuance schedule. It does not restore the original owner’s access.

6.3 Long-term mining resources

A funded pool supplements fees and ordinary issuance. It offers a way to allocate existing units toward future mining compensation.

6.4 Retirement of old state

Outputs cannot remain in the UTXO set indefinitely merely because nobody spends them. Expiry removes old surviving outputs.

This bounds output age, rather than total UTXO size. High output creation can still produce substantial state.

6.5 Public, uniform processing

No administrator must determine whether a key is lost or an owner has died. Nodes apply the same creation-height and unspent-status conditions.

Future miners can receive previously inaccessible amounts through successful block production. This provides an acquisition path and a possible resource for renewed participation, subject to mining competition and demand.

---

7. Cryptographic Retirement

If a consensus change stops creation of outputs using an obsolete signature or script mechanism, existing outputs of that type can be migrated or eventually expire.

This establishes a finite horizon after which no live outputs remain protected by that mechanism.

Two limits remain:

- A mechanism compromised before expiry still requires timely security action.
- Independent historical replay still requires the validation rules applicable to past blocks.

Retirement from live state does not imply that historical validation support can be discarded.

---

8. Storage and Historical Verification

Expiry, pruning, and full historical replay serve different purposes.

We support two sources of expiry candidates:

Mode	Candidate source	Storage consequence
Block mode — default for new data directories	Original blocks that created the outputs	Retains the expiry window plus a reorganization buffer
Index mode	A height index derived from live UTXOs	Supports pruning without retaining origin blocks for the full expiry window

Existing databases retain their recorded mode unless explicitly changed through the required rebuild procedure.

In default block mode, approximately the first century of operation requires retaining the full block history, even when pruning is enabled. During this period, a pruning target does not impose a disk-usage ceiling.

Index mode trades additional live-state indexing and writes for direct expiry lookup and practical pruning. Both modes enforce the same expiry and pool rules.

A node continuing from an independently validated state needs the data required by its selected mode and recovery policy. A node replaying from genesis needs the full historical block record and historical consensus rules.

Sources: node operation and defaults and storage design.

---

9. Ownership Trade-off: Is Expiry Confiscation?

An owner who retains a valid private key but fails to renew loses the ability to spend the expired output. Its amount becomes available for future miners.

This is a substantive restriction on ownership. Describing it as recycling does not remove that consequence.

Our position is that the restriction is public from output creation, applies uniformly, and provides a long renewal period. Owners can retain control by creating fresh outputs under their own keys without transferring ownership to another person.

Public disclosure does not make the condition acceptable to everyone. A holder who requires indefinite spendability without any future on-chain action may reasonably prefer a different monetary system.

Resatoshi exchanges that indefinite guarantee for finite unattended state and the reuse of expired amounts.

---

10. Security and Economic Limitations

10.1 Renewal censorship

A miner may exclude a renewal transaction near its deadline. If no competing block includes it in time, the output expires and increases the recycling pool.

This can create an incentive conflict, particularly for miners with substantial hash power.

The payout cap spreads benefits over future blocks and can delay their collection. It does not eliminate the incentive: the responsible miner may collect part of the resulting rewards over time.

Renewing years or decades before expiry reduces reliance on inclusion within a narrow deadline. Sustained censorship nevertheless remains a security concern.

The incentive depends on existing pool funding, hash share, competing miners, forgone fees, and when the extra pool balance would affect payouts.

10.2 Mining interruption

Expiry advances with block height. If block production stops, expiry and payouts also stop.

A recycling pool does not automatically restart a stopped chain.

10.3 Economic security

RST-denominated rewards do not establish purchasing power or the hash power they will attract. Security depends on economic conditions and participation, as well as the reward schedule.

10.4 Concentrated expiry workload

Many outputs may expire at one height, concentrating validation and storage work.

The 1 RST cap limits payout speed, not expiry-processing cost.

10.5 Timelocks

An output whose spending conditions remain locked beyond its expiry boundary can expire before it becomes spendable.

Long-term scripts must account for both their spending conditions and the output lifetime.

10.6 Custody and inheritance

Keys alone are insufficient for indefinite unattended custody. Owners and heirs must also retain knowledge of expiry and complete renewal in time.

10.7 Depletion and concentration

The pool can empty, and rewards follow successful block production. Recycling provides neither perpetual funding nor equal distribution.

---

11. Upstream Maintenance

We use Bitcoin Core upstream as a common reference for maintenance. Public source and change history allow contributors to review improvements, propose updates, and continue development without depending on the original developer.

A common reference supports consistency, while explicit fork changes show successors which rules must be preserved. Adoption remains a decision for network participants.

The project has integrated Bitcoin Core 32.1. At source commit 89089ee5, comparison against the official 32.1 source archive gives:

Comparison scope	Changed files	Added lines	Removed lines
All non-vendor files	342	24,197	3,352
Explicit selection of consensus-related files	9	609	104

The total includes documentation, validation evidence, branding, tests, and other project material.

The selected subset covers chain and consensus parameters, difficulty adjustment, recycling, block validation, and undo records. It is a defined comparison scope, not a complete measurement of every consensus-sensitive dependency or a security metric.

Upstream changes affecting UTXO handling, rewards, validation, reorganizations, or snapshots require compatibility review even when they merge without textual conflicts.

References: source snapshot and upstream integration record.

---

12. Implementation and Validation

Rule or mechanism	Implementation
5,256,000-block lifetime	src/consensus/params.h
Transactions before expiry	src/validation.cpp, src/node/recycle.cpp
Same-block expiry funding	src/node/recycle.cpp
Maximum 1 RST recycling reward	src/node/recycle.h, src/node/recycle.cpp
Optional claims; unclaimed amounts retained	src/node/recycle.cpp
Per-block expiry and pool undo	src/undo.h, src/node/recycle.cpp
Genesis-anchored ASERT	src/pow.cpp, src/pow/asert.cpp
Block and index expiry modes	src/node/recycle.cpp
Index diagnosis	checkrecycleindex in src/rpc/blockchain.cpp
Ordinary issuance schedule	src/validation.cpp, src/kernel/chainparams.cpp

Our Bitcoin Core 32.1 integration validation replayed mainnet history through height 13,600 using the previous and updated implementations in both expiry modes. All four runs agreed on block identity, UTXO state, and pool balance.

Because mainnet has not reached expiry age, expiry behavior is tested separately with accelerated regtest lifetimes. These tests cover renewal, nonzero pools, reorganizations, restart, and reindexing.

The full-block comparison includes 50,100 outputs expiring at one height. Testing provides evidence for the exercised cases; it does not establish correctness for every possible future state.

Source: validation procedures and results.

---

13. Conclusion

Resatoshi gives unspent outputs a finite lifetime and redirects expired amounts into capped future mining rewards.

Its central trade-off is explicit: holders must renew outputs to retain control. In exchange, inaccessible amounts can return to circulation, old unused state can retire, and existing units can support future mining without extending issuance.

The mechanism begins producing rewards only after approximately a century. Its long-term effect will depend on renewal behavior, pool funding, market demand, mining participation, and correct implementation.