Blackcoin

Quantum Quasar

Protocol V4

A Post-Quantum, Participation-First Evolution of Blackcoin

Technical White Paper

Quantum Quasar Developers

Version 30.1.1 - July 2026

https://github.com/Blackcoin-Dev/Blackcoin

Contents

Abstract

Blackcoin was one of the first pure Proof-of-Stake (PoS) cryptocurrencies, launched in 2014. Proof-of-Stake secures a network with coins rather than energy, which makes the long-term integrity of the chain depend on two things: the secrecy of the signing keys that protect stake, and the willingness of coin holders to actually *use* those keys to mint blocks. Both of these are now under pressure. The arrival of cryptographically relevant quantum computers threatens every coin that is protected only by elliptic-curve (ECDSA/Schnorr) signatures, and the long tail of dormant, never-staked coins slowly erodes the active security budget of any PoS chain.

Quantum Quasar (Protocol V4) is Blackcoin's answer to both problems at once. It introduces a NIST-standardized post-quantum signature scheme (ML-DSA-44) as a first-class, consensus-enforced spending path; a time-boxed, deterministic migration from legacy elliptic-curve outputs to quantum-safe outputs; and a set of participation mechanics, the Gold Rush reward epoch, liveness demurrage, tiered and cold quantum staking, and a bounded legacy lockout, that are explicitly designed to convert passive holding into active network security **without punishing the holder who is willing to participate**.

This paper documents the V4 protocol in exhaustive detail: every consensus constant, every phase boundary, every reward formula, and the exact wallet workflows and RPC commands a user needs. All numbers in this document are taken directly from the v30.1.1 release source and are annotated with the file that defines them.

1. Design Philosophy: Participation Over Passivity

Every design decision in Quantum Quasar answers a single question: *does this reward the people who actively secure the network, and does it give everyone else a clear, fair, fully-rewarded path to join them?*

Traditional PoS has a free-rider problem. A large holder can leave coins in a cold wallet for years, contribute nothing to block production, and yet retain the full economic weight of those coins. Meanwhile the holders who run nodes and mint blocks carry the real security burden. Over time this concentrates security into fewer hands and leaves the chain vulnerable, because so much of the coin supply is "asleep."

Quantum Quasar is built on the opposite premise: **the network should reward participation and slowly reclaim the influence of pure inactivity, while making participation cheap, automatic, and profitable for anyone willing to do it.** The three pillars below all serve that goal.

- **Full rewards for HODLers who help secure.** The Gold Rush epoch pays a large, deterministic bonus emission to holders, but only through the act of staking (PoS) or mining (PoW). Holding qualifies you; participating pays you. A holder who simply keeps their node online and staking earns their full share.
- **Liveness demurrage instead of dead weight.** After the migration era, quantum holdings that remain inactive for more than six months begin a slow, capped decay. Decayed principal is permanently burned when spent; it is never added to transaction fees or paid to a miner or staker. The point is to ensure that keys are alive and that the security weight of the supply reflects who is actually present. For an eligible direct or tiered v16 holding, the wallet can attempt a low-fee liveness attestation when staking is enabled, the wallet is normally unlocked, and a safe fee input is available. A cold-stake output is also subject to the activity clock; a successful coinstake spends and recreates it, resetting that clock.
- **A bounded, well-signposted quantum migration.** Legacy elliptic-curve outputs are the network's quantum attack surface. Gold Rush keeps ordinary quantum funding disabled so the base chain remains legacy-compatible. It is followed by an **18-month Migration phase** in which every holder can move coins into quantum-safe addresses with a one-click wallet operation. Final Lockout then closes the legacy spending path. Legacy value stays spendable for the full Gold Rush-plus-Migration schedule, but the

migration transaction itself must be made during Migration.

The result is a network that trends toward *more* active nodes, *more* distributed security, and *more* engaged holders over time, the opposite of the slow ossification that afflicts passive-holding chains.

2. The V4 Timeline: Four Phases

Protocol V4 uses a complete, height-authoritative lifecycle on Blackcoin mainnet. MTP anchors remain as nominal forecasts and for isolated compatibility tests, but timestamp movement cannot advance, delay, skip, or reverse a mainnet phase boundary.

Phase	Mainnet heights	Target duration	Legacy spend?	v14/v16 funding and spending?	Gold Rush rewards?
Legacy	through 5,949,999	-	Yes	No	No
Gold Rush	5,950,000-6,192,999	243,000 blocks (~180 days)	Yes	No; shadow credits are locked	Yes
Migration	6,193,000-6,921,999	729,000 blocks (~540 days)	Yes	Yes	No
Final Lockout	from 6,922,000	permanent	No	Yes	No

The quantum column refers only to authenticated witness-v14 and witness-v16 paths. Witness v15 has no supported funding or spending workflow in any v30.1.1 phase; consensus rejects v15 outputs and spends from Migration onward.

Exact schedule (mainnet)

The consensus boundaries are defined in `src/shadow_schedule.cpp`, `src/shadow.h`, `src/consensus/params.h`, and `src/kernel/chainparams.cpp`.

- **V4 and Gold Rush begin:** height **5,950,000**.
- **Last Gold Rush block:** height **6,192,999**. Migration begins at **6,193,000**.
- **Last Migration block:** height **6,921,999**. Final Lockout and automatic demurrage begin together at **6,922,000**.

The retained time anchors are non-authoritative mainnet forecasts:

- `QUANTUM_QUASAR_MAINNET_V4_TIME = 1783835299` is **2026-07-12 05:48:19 UTC**.
- Adding the 180-day nominal Gold Rush duration gives `1799387299`, **2027-01-08 05:48:19 UTC**.
- Adding the 540-day nominal Migration duration gives `1846043299`, **2028-07-01 05:48:19 UTC**.

Phase membership is computed by `GetQuantumLifecycleState(nTime, nHeight)` (`src/consensus/params.h`). Because Blackcoin mainnet targets a **64-second block time**:

- **1,350 blocks/day**, **40,500 blocks/month** (30-day month), **~493,000 blocks/year**.

Legacy elliptic-curve coins remain spendable for Gold Rush and Migration, approximately 720 target days in total. Ordinary quantum funding is deliberately off during the first 180 target days, so holders perform `migratetoquantum` during the following 540-day Migration phase. Final Lockout closes the legacy path at height 6,922,000.

3. Post-Quantum Cryptography in Blackcoin

3.1 Why elliptic curves are the problem

Blackcoin's legacy outputs are protected by ECDSA over secp256k1. A sufficiently large quantum computer running Shor's algorithm can recover a private key from a public key. Any coin whose public key is exposed on-chain, which includes every P2PK coinstake output Blackcoin has ever produced, and any address that has spent before, is at risk the moment such a machine exists. PoS chains are especially exposed because staking continuously publishes public keys.

3.2 ML-DSA-44: the quantum-safe signing path

V4 introduces **ML-DSA-44** (Module-Lattice Digital Signature Algorithm, the NIST FIPS 204 standardization of CRYSTALS-Dilithium at security level 2) as a native, consensus-verified signature scheme, provided by liboqs 0.15.0 and wrapped in `src/crypto/mldsa.h`. ML-DSA's security rests on the hardness of module lattice problems, which are not known to be broken by quantum algorithms.

Quantity	Size
Public key	**1,312 bytes**
Secret key	2,560 bytes
Signature	**2,420 bytes**

These are large compared to a 33-byte ECDSA public key and ~72-byte signature, the price of quantum resistance, which is why V4 places quantum data in the witness (where it is discounted) and uses commitment-based addresses so that the large key is only revealed at spend time.

3.3 New witness versions and address types

V4 defines three new SegWit witness versions. Direct v14/v16 programs and the reserved v15 shape are 32 bytes; tiered v14/v16 programs are 40 bytes (`src/consensus/quantum_witness.h`, `src/addresstype.h`). All use the mainnet Bech32 human-readable prefix `blk` (`src/kernel/chainparams.cpp`).

Witness v	Program	Purpose
v16	32-byte direct or 40-byte tiered program	**Quantum migration / tiered staking** output: an ML-DSA-protected home for migrated coins. Subject to demurrage.
v15	32-byte commitment	**Reserved EUTXO shape**: disabled/frozen in v30.1.1 because it has no quantum ownership authorization.
v14	32-byte direct or 40-byte tiered program	**Quantum cold-stake**: owner/staker-separated delegation output, subject to inactivity demurrage.

For authenticated v14 and v16 paths, the base program carries a 32-byte commitment. A tiered program prepends eight bytes of state to a 32-byte commitment. The bulky ML-DSA public key and signature are supplied in the witness at spend time. The reserved v15 commitment does not commit to an ML-DSA owner; this is why v30.1.1 must not fund or spend it.

3.4 Self-test on startup

The node performs an ML-DSA Known-Answer-Test at startup to verify that liboqs is linked correctly and produces standard-conformant signatures before it will participate in consensus. A build that cannot reproduce the ML-DSA KAT refuses to run, guaranteeing that every V4 node validates quantum signatures identically.

4. The Gold Rush Reward Epoch

The Gold Rush is a six-month, deterministic bonus-emission event that launches V4. Its purpose is to **reward existing holders for participating in securing the network at the exact moment the network needs maximum participation**, the transition into the quantum era. It is not an airdrop to passive wallets; it pays holders through the act of staking and mining.

4.1 The whitelist snapshot

At height **5,945,000** (`SHADOW_WHITELIST_HEIGHT`, `src/shadow_schedule.cpp`), the node builds a **deterministic snapshot** of every script holding at least **10,000 BLK aggregate** (`SHADOW_WHITELIST_MIN_BALANCE`, `src/shadow.h`). The snapshot is computed once from the UTXO set at that exact height and is read-only thereafter, so every node derives an identical whitelist.

- Balances are aggregated **per canonical script.** P2PK stake scripts are folded to their P2PKH identity via `CanonicalizeLegacyStakeScript()` so that a holder who staked with raw-pubkey outputs and a holder who used address outputs are treated as one account.
- The snapshot happens **5,000 blocks (~ 3.7 days) before** Gold Rush rewards begin at height 5,950,000, giving the network a clean, pre-announced eligibility set that cannot be gamed after the fact.

> **Why 10,000 BLK?** The threshold defines the set of accounts large enough to be meaningful security participants during the transition. Being whitelisted is necessary to earn Gold Rush *credits*, but it does not by itself pay anything, the holder still has to show up and stake.

4.2 The reward schedule

Each Gold Rush block accrues a base reward, `ShadowBaseReward(height)` (`src/shadow.cpp`), over the window from height 5,950,000 to **6,192,999** (`SHADOW_REWARD_END_HEIGHT`), a span of `SHADOW_GOLD_RUSH_BLOCKS = (180 x 24 x 60 x 60) / 64 = 243,000` blocks.

Phase 1 (heights 5,950,000-6,187,599): a halving curve.

```
reward = (580 BLK) >> (blocks_since_start / 43,200)
```

The reward starts at **580 BLK/block** and halves every `SHADOW_HALVING_INTERVAL_BLOCKS = 43,200` blocks (~ 30 days):

Month	Height range	Reward/block
1	5,950,000-5,993,199	580 BLK
2	5,993,200-6,036,399	290 BLK
3	6,036,400-6,079,599	145 BLK
4	6,079,600-6,122,799	72 BLK

Month	Height range	Reward/block
5	6,122,800-6,165,999	36 BLK
6 (part)	6,166,000-6,187,599	18 BLK

Phase 2 (heights 6,187,600-6,192,999): a fixed tail.

```
reward = 463 BLK/block
```

The final ~5,400 blocks pay a flat **463 BLK/block**. The two-phase shape is calibrated so that total Gold Rush accrual lands at the hard cap `SHADOW_MAX_EMISSION = 51,437,700 BLK` (`src/shadow.h`); the pool cannot over-issue beyond this cap regardless of block timing.

4.3 The PoS / PoW split

Each block's base reward is divided **evenly** between two independent reward pools (`src/shadow.cpp`):

```
pos_pool_reward = reward - reward/2      (50%)
pow_pool_reward = reward/2               (50%)
```

- **The Proof-of-Stake half** rewards Blackcoin's native stakers. This is the primary, energy-free path and the one most holders will use: keep your node online, stake, and earn.
- **The Proof-of-Work half** opens a parallel, opt-in participation lane using a deliberately *CPU-friendly, memory-light* Argon2id puzzle (`SHADOW_ARGON2_TIME_COST = 1`, `SHADOW_ARGON2_MEMORY_KIB = 1024` (1 MiB), `SHADOW_ARGON2_LANES = 1`), so ordinary community members, not just ASIC farms, can contribute and claim.

Rewards accumulate into pools and are drawn by **claims**, not paid blindly to whoever found a block, which lets both PoS and PoW participants collect their fair share over the epoch.

4.4 Qualifying and claiming: QQSIGNAL and QQSPROOF

Gold Rush participation is expressed through two on-chain, fee-paying control transactions carried in OP_RETURN outputs:

- **QQSIGNAL (Proof-of-Stake side).** A whitelisted holder who has produced a recent PoS block signals eligibility by broadcasting a QQSIGNAL that references their recent solve. "Recent" is defined by the solver-activity window `SHADOW_SOLVER_ACTIVITY_SECONDS = 14 days` (`SHADOW_SOLVER_ACTIVITY_WINDOW = 18,900 blocks`, `src/shadow.h`). In other words: stake a block, and you have a 14-day window to signal and be credited from the PoS pool. This is the mechanism that ties reward to genuine, ongoing participation rather than to a one-time balance.
- **QQSPROOF (Proof-of-Work side).** A miner grinds an Argon2id proof (magic `QQSPROOF`, `src/shadow.cpp`) against the target difficulty, a 12-bit base, ASERT- retargeted every 64 blocks within a [10-bit, 18-bit] band, and submits it in a claim transaction that is validated before mempool acceptance. Already-mined blocks through 5,993,199 retain the v30.1.0 first-valid-claim allocation. At the first scheduled halving, height 5,993,200, v30.1.1 begins the QQP3 canonical rank-v1 rule, ranking competing candidates independently of transaction order and evaluating at most 64 Argon2 proofs. QQP3 binds each new proof to its intended height and parent and permits fee-only reimbursement for 64 later blocks on the same branch. The lowest-ranked valid current-origin claim receives the fixed PoW pool after every current loser and eligible late claimant is reimbursed its actual base fee, capped at 0.01 BLK. A late-only block leaves the unreimbursed pool accumulated. At most 0.63 BLK can be reimbursed alongside a winner. Invalid,

malformed, expired, off-branch, and excess claims receive nothing.

QQP4 additionally binds the exact single legacy fee-input outpoint. It has a separate consensus activation and is disabled on mainnet in v30.1.1. Readiness or version-bit signalling cannot activate QQP4. Any future QQP4 release must publish an explicit activation height and a tested transition for QQP3 claims that are still inside their late-inclusion window.

The wallet can automate both when the corresponding staking/mining mode and signing prerequisites are satisfied. See §9 for the exact RPCs (`sendshadowsignal`, `sendshadowpowclaim`, `setpowmining`, `getgoldrushinfo`).

A wallet-authored `QQSPROOF` that leaves the local mempool remains unresolved: another peer can retain and later confirm the base-valid transaction. The wallet therefore quarantines the claim, reserves its exact fee input, refuses generic abandonment, and pauses its built-in miner instead of creating a second claim. `getpowmininginfo` reports the live, quarantined, and total unresolved claim counts.

The consent-only `createshadowpowclaimresolution` RPC can preview an exact-input, same-script conflict. Its default dry run does not sign. A signing request requires a normally unlocked wallet and explicit acknowledgement of the fee and conflict risk, but the RPC still never broadcasts; the operator must review the result and separately call `sendrawtransaction`. A confirmed resolution pays its base-chain fee without shadow reimbursement. Peers retaining the original may reject the conflict, confirmation is not guaranteed, and the input remains reserved until one transaction confirms. After an explicit broadcast and wallet recognition, ordinary wallet rebroadcast may continue across restart; this is downstream of the prior broadcast decision.

5. Quantum Migration and the Legacy Lockout

5.1 The problem being solved

Every legacy Blackcoin output is protected by ECDSA and therefore represents standing quantum risk to the whole network, not just to its owner. Leaving that risk open forever would mean the chain's security never actually improves, no matter how good the new cryptography is. V4 therefore treats migration as a **network-wide, time-boxed public project** with a clear deadline.

5.2 The migration path

Any holder moves coins to safety with a single wallet action, `migratetoquantum` (see §9), which sweeps spendable legacy (non-quantum) outputs into a fresh **wallet-backed quantum migration (witness v16)** address. The destination ML-DSA key is generated and **written to the wallet database before any funds move**, and the call refuses to proceed unless that key is confirmed stored, so a migration can never send coins to a key the wallet does not hold.

> **Critical backup note.** ML-DSA keys are *not* derived from the wallet's HD seed. After creating a migration address you **must back up the wallet again**, or a restore from an older backup will not recover the migrated funds. The wallet and this paper both flag this at every step.

During Gold Rush, wallets can create and back up quantum addresses and can dry-run migration planning with an existing wallet-backed address. They cannot fund or spend ordinary v14/v16 outputs. `migratetoquantum` becomes actionable at Migration height 6,193,000 and remains available through height 6,921,999. Gold Rush

reward credits are separate authenticated synthetic outputs that remain phase-locked until Gold Rush ends and normal maturity is satisfied.

5.3 The lockout, and why it is a feature

At Final Lockout, **height 6,922,000**, the consensus rule `IsQuantumFinalLockout(nTime, nHeight)` (`src/consensus/params.h`, enforced in `src/validation.cpp`) becomes true, and the script flag `SCRIPT_VERIFY_LEGACY_ECDSA_LOCKOUT` (`src/script/interpreter.cpp`) causes **all legacy ECDSA-signed spends to be permanently rejected** with `legacy-spend-disabled`. Exact authenticated v14 and v16 paths remain enabled. Witness-v15 funding and spending remain rejected with the dedicated EUTXO-disabled rules.

This is deliberately framed as a positive:

- It converts an **unbounded, permanent, network-wide** quantum vulnerability into a **finite, scheduled, individually-avoidable** one. After the deadline, the set of quantum-vulnerable coins can only shrink, never grow.
- The schedule is **generous and loud:** about six months of Gold Rush preparation plus an 18-month Migration phase, with the exact height deadline visible in the wallet, network status RPCs, and this document.
- It is **individually avoidable in one click.** A holder who migrates during the 540-day Migration phase retains spendable control.
- It **strengthens every remaining coin.** Once legacy spends are closed, the entire active supply is quantum-safe, which raises the security floor for everyone who participated.

The lockout is the mechanism that guarantees the migration actually completes, rather than dragging on forever with a permanently vulnerable dormant tail.

6. Demurrage: Liveness as a Public Good

Demurrage is the most misunderstood, and most important, participation mechanic in V4. This section states plainly what it is and what it is not.

It is a liveness rule for quantum holdings. A timely direct-key attestation or an activity spend refreshes the clock. Delegation alone does not. Demurrage applies to eligible quantum outputs left inactive for **more than six months**.

6.1 Exactly which coins are subject

Evaluated per output by `EvaluateDemurrage` (`src/consensus/demurrage.cpp`). The classification, in order:

1. **Demurrage not yet active** -> exempt. Demurrage cannot begin before the migration era (mainnet's height-authoritative schedule ends Migration at block 6,921,999 and activates Final Lockout and demurrage at block 6,922,000).
2. **Explicitly configured exempt scripts** -> exempt. Mainnet currently configures none.
3. **Non-quantum (legacy and everything else) outputs** -> exempt (`"non_quantum"`). Legacy coins are governed by the lockout, not demurrage.
4. **Quantum migration/tiered (v16) and cold-stake (v14) outputs** are subject, but only if inactive beyond the grace period. Eligible direct/tiered v16 keys can use attestations; cold-stake state refreshes through a

successful spend/recreation.

5. **Historical v15-shaped outputs** can be classified by the accounting evaluator, but v30.1.1 independently rejects their funding and spending. They have no supported liveness or recreation path, and their metadata is inspection-only.

Thus cold delegation is not a permanent-value shelter. A cold-stake output that never successfully stakes or moves follows the same inactivity curve.

6.2 The inactivity clock and the grace period

For a subject output, the node computes:

```
effective_last_active = max(coin_creation_height,
                            demurrage_activation_height,
                            latest_attestation_height)
inactive_blocks = spend_height - effective_last_active
```

If `inactive_blocks <= DEMURRAGE_GRACE_BLOCKS` (**243,000 blocks = 6 months**), the output is **exempt**, full value, no decay (`"young"` if freshly created/moved, `"attested"` if kept alive by a qualifying attestation). Creating, moving, or receiving an output resets its clock. A qualifying attestation resets the clock only for an eligible direct/tiered v16 key.

6.3 The decay curve

Past the grace period, value decays **quadratically** to zero over an 18-month window (`DemurrageRemainingPpm`, `src/consensus/demurrage.cpp`):

```
grace_blocks = 243,000                (6 months)
zero_blocks  = 972,000                (24 months)
decay_window = 729,000                (18 months)

elapsed      = inactive_blocks - grace_blocks
t            = (elapsed / 729,000) x 1,000,000       (parts per million)
remaining    = 1,000,000 - (t² / 1,000,000)          (ppm of value retained)
```

The value burned when such a coin is spent is `nominal - effective`. Consensus recognizes only the effective value as input principal. The transaction fee is then `effective inputs - outputs`, so the burned remainder is not a fee and is never paid to the block producer. A coinstake is governed by the same rule: only effective principal is returned, and its reward remains limited to the ordinary PoS subsidy plus explicit fees.

Decay table (a subject output with no qualifying attestation, spend/recreation, or successful coinstake):

Months inactive	Value retained
<= 6 (grace)	**100.0%**
9	97.2%
12	88.9%
15	75.0%
18	55.6%
21	30.6%
24	**0.0%** (locked)

Note the shape: the first year barely moves (the curve is quadratic, so early decay is tiny), and the losses only become material deep into the second year of *total* neglect. This is by design, it gives even a careless holder a very long runway, while ensuring that genuinely dead coins eventually stop counting as security weight.

6.4 Keeping eligible holdings at 100%

There are three ways to keep a quantum holding at full value. Which one applies depends on the output type and wallet state:

- **Wallet-assisted liveness attestation.** A demurrage attestation is a zero-value, fee-only transaction carrying an ML-DSA signature (`senddemurrageattestation`) that resets an eligible direct or tiered v16 key's clock. An attestation is valid for **6 months** (`DEMURRAGE_ATTEST_VALIDITY_BLOCKS = 243,000`), and the wallet attempts one at the **3-month** mark (`DEMURRAGE_AUTO_ATTEST_BLOCKS = 121,500`) only while staking is enabled, the private-key wallet is normally unlocked rather than staking-only, and a safe spendable fee input is available. Capacity, construction, or broadcast failures can defer an attempt; merely leaving a wallet online is not a guarantee.
- **Active cold staking.** Cold-stake (v14) outputs remain subject to demurrage. Each successful coinstake realizes any accrued burn, returns only effective principal plus the ordinary reward, recreates the output, and resets its activity clock. Delegation by itself is not an exemption.
- **Any ordinary use.** Moving, consolidating, or spending resets the clock as a side effect.

The `getdemurragewalletinfo` RPC reports, per output, the nominal amount, the current effective (post-decay) amount, the value that would be burned if spent now, whether an attestation is due, and whether the output is locked. `sweepdemurragedecay` spends outputs that are decaying but still have positive effective value, realizes the burn, and moves the remainder minus the explicit fee to a fresh quantum address. Outputs at zero effective value are permanently locked and are skipped. The GUI surfaces the same information and can request an attestation for an eligible selected address; normal unlock, key, fee, and broadcast requirements still apply.

In short, timely participation preserves principal. Any decay realized by a spend is destroyed, not redistributed.

7. Quantum Staking: Tiered, Cold, and Pooled

V4 gives quantum coins a rich set of ways to participate and earn rewards. Participation also refreshes activity when it spends and recreates the relevant output.

7.1 Tiered self-staking

A holder can bond quantum coins into a **tiered self-staking** output that encodes a lock schedule directly in the witness program (`QuantumStakeTierProgram`, `src/consensus/quantum_witness.h`):

- **State machine:** `BONDED` -> (initiate unbonding) -> `UNBONDING` until `unlock_height` -> spendable.
- **Unbonding period** is chosen by the holder (in blocks; at 64 s/block, e.g. 40,500 blocks = 30 days) and is stored in the program itself, so the lock is consensus-visible.
- **RPCs:** `fundquantumstakeaddress` (bond), `withdrawquantumstakeaddress` (unbond or withdraw matured funds, the call is state-aware), `getquantumstakeaddressinfo`, `listquantumstakeoutputs`.

Longer, more committed locks express stronger participation and are ranked accordingly in the pool logic below.

7.2 Cold staking: separate the owner key from the staking key

Cold staking (witness v14) splits control into an **owner key** (can withdraw the principal) and a **staker key** (can only mint blocks with the coins), hashed together under the domain tag *"Quantum Quasar Cold Stake v2"* (`src/consensus/quantum_witness.cpp`). This lets a holder keep their principal in cold storage while a hot node, their own or a trusted operator's, stakes on their behalf.

- The delegated principal is always **owner-controlled**; the operator can never move it.
- Cold-stake outputs are subject to the same inactivity schedule. A successful coinstake returns effective principal plus reward and creates a fresh output, but delegation alone does not freeze the clock.
- **RPCs:** `fundquantumcoldstakeaddress`, `withdrawquantumcoldstakeaddress`, `getquantumcoldstakebalance`.

7.3 Operator bonds and the per-pool cap

Operators who stake on behalf of others post a **30-day operator bond** (40,500 blocks, `src/wallet/rpc/staking.cpp`; pool logic in `src/node/quantum_pool.h`), which registers a verified commitment other participants can see. To keep the network decentralized, a **wallet/policy per-pool cap of 20%** (`QUANTUM_POOL_CAP_BPS = 2000`) discourages any single operator from accumulating an outsized share of delegated stake. The cap is a *wallet and delegation policy*, not a consensus rule, it steers new delegations without changing block validity, so it can be tuned by the community without a fork.

- **RPCs:** `fundquantumoperatorbond`, `withdrawquantumoperatorbond`, `getquantumoperatorbondinfo`, `getwalletquantumpoolinfo` (verified value, share in basis points, valid/invalid claim counts, over-cap flag per operator).

7.4 Autonomous redelegation

An unlocked owner wallet can use the redelegation engine (`src/wallet/redelegation.h`) to move an eligible delegation after its current operator has produced no observed wins for the policy interval and a meaningfully better verified target is available. A pool exceeding the policy cap does not itself trigger redelegation.

- **Trigger:** `6 x expected_interval_blocks`, clamped to **300-4,050 blocks**, followed by deterministic per-delegation jitter of up to **1,350 blocks**.
- **Dampers:** attempts and successes have a **1,350-block** rate limit. A separate **1,350-block** probation period runs from the current delegation's activation height; it is not a probation period created by a failed attempt.
- **Ranking:** verified non-current operators are ranked first by observed liveness and then by smaller pool share. Over-cap targets are removed when an under-cap alternative exists. If every verified candidate is over the cap, the bootstrap fallback permits those candidates and reports the projection.
- **Execution prerequisites:** automatic mode must be enabled, the owner wallet must be normally unlocked with private keys, and the delegation must be safe and owner-spendable. A missing target or transaction failure leaves the delegation unchanged.
- **RPCs:** `getquantumredelegationinfo` (dry-run/status), `redelegatequantumcoldstake` (manual, with cap enforcement and a dry-run mode).

The policy can reduce maintenance for an unlocked owner wallet, but it is not a consensus guarantee of liveness or pool distribution and does not remove the need to monitor failed or deferred attempts.

8. Reserved v15 EUTXO Design and RGB

V4 includes an operational RGB commitment path and retains a reserved EUTXO encoding for inspection and future design work. Those two states must not be treated as equally enabled.

8.1 EUTXO witness v15 is frozen in v30.1.1

The reserved **EUTXO-shaped output** (witness v15, `src/addresstype.h`) commits to a *datum* and *validator script* as `SHA256("Quantum Quasar EUTXO v1" || SHA256(datum) || SHA256(validator))` (`src/script/solver.cpp`). That commitment does not authenticate a quantum owner. A validator-only spend design would therefore permit value movement without the ML-DSA ownership guarantee required by the migration.

v30.1.1 sets `EUTXO_ENABLED` to false. Wallet and raw-transaction construction reject v15 outputs, the `createeutxospend` and `createeutxotransition` RPCs intentionally return a disabled error, and post-Gold-Rush consensus rejects both v15 funding and v15 spends. `decodeeutxospend` and `verifyeutxospend` remain inspection tools only; `importeutxostate` and `listeutxostates` only persist or report metadata and cannot make a v15 output spendable. **Do not send BLK to a witness-v15 address in v30.1.1.**

8.2 RGB, client-side fixed-supply assets

An **RGB commitment** anchors a client-side-validated asset state transition in a zero-cost OP_RETURN (`OP_RETURN <RGB1> <32-byte state hash>`, `src/script/solver.cpp`). The heavy asset data lives off-chain and is validated by clients against the on-chain commitment chain, so Blackcoin can carry fixed-supply tokens and assets without bloating the base layer.

- **Tooling:** `creatergbtransfer`, `acceptrgbconsignment`, `exportrgbconsignment`, `importrgbcontract`, `importrgbassignment`, `listrgbassets`, plus raw `decodergbcommitment` / `verifyrgbconsignment`.

9. Full Wallet and RPC Reference

Every command below is available from `blackcoin-cli` and the Qt debug console. This is the complete Quantum-Quasar-specific surface added on top of the standard Bitcoin/Blackcoin RPC set.

9.1 Chain and schedule

RPC	Purpose
`getquantumquasarinfo`	V4 phase, activation/Gold-Rush/deadline times, next-block phase
`getgoldrushstate`	Chain-level Gold Rush (Shadow Network) state
`getgoldrushinfo`	Gold Rush pools (PoS/PoW amounts), solver counts, wallet qualification
`getcirculatingsupply`	Demurrage-adjusted circulating supply; guarded full scan with explicit one-call consent outside its reviewed envelope
`getshadowresourceinfo`	Inspect optional supply-scan qualification, bounds, warnings, and progress

RPC	Purpose
`abortcirculatingsupplyscan`	Request cooperative cancellation of the active full-supply scan
`getquantumpoolinfo`	Non-consensus quantum cold-stake pool registry

9.2 Staking, Gold Rush, and mining

RPC	Purpose
`getstakinginfo` / `staking`	Read / start-stop legacy PoS staking
`reservebalance`	Reserve coins from staking/spending
`getstakingdonationinfo` / `setstakingdonation`	Read / set staking-reward donation percentage
`checkkernel`	Test whether an input is a valid PoS kernel now
`sendshadowsignal`	Broadcast a QQSIGNAL for a recent PoS solve (Gold Rush PoS credit)
`sendshadowpowclaim`	Grind and submit a QQSPROOF Argon2id PoW claim
`setpowmining` / `getpowmininginfo`	Control / inspect the in-process Argon2id miner
`createshadowpowclaimresolution`	Preview or, after explicit acknowledgement, sign but never broadcast an exact-input conflict for one quarantined wallet claim
`optimizeutxoset`	Rebuild the UTXO set into equal outputs to maximize PoS yield

9.3 Quantum addresses and migration

RPC	Purpose
`getnewquantumaddress` / `listquantumaddresses`	Create / list wallet-backed ML-DSA migration addresses
`createquantumkey`	Fail-closed deprecated stub; v30.1.1 never generates unstored raw key material
`createquantummigrationaddress`	Encode a migration commitment from an existing ML-DSA public key
`dumpquantumkey`	Expert-only export from the selected normally unlocked wallet; disabled unless `-allowunsafequantumkeyrpc`; staking-only unlock is rejected
`importquantumkey`	Import an ML-DSA key into the selected wallet
`migratetoquantum`	Sweep legacy coins into a quantum migration address
`migrategoldrushrewards`	Move Gold Rush reward outputs to a fresh quantum address
`getmigrationstatus`	Migration progress, eligible legacy amount, deadline countdown, advice

v30.1.0 exposed raw unstored key generation through `createquantumkey`. v30.1.1 retains that RPC name only as a fail-closed migration stub. New keys are created through wallet-backed RPCs and must be covered by a verified backup. `-allowunsafequantumkeyrpc` enables only `dumpquantumkey`; it does not make the node offline or restrict RPC access, so exports belong in a separately isolated offline process.

9.4 Quantum staking, cold staking, operator bonds

RPC	Purpose
`getnewquantumstakeaddress`	New tiered self-staking address
`fundquantumstakeaddress` / `withdrawquantumstakeaddress`	Bond / unbond-withdraw tiered stake
`getquantumstakeaddressinfo` / `listquantumstakeoutputs`	Inspect tiered stake state and outputs
`getnewquantumcoldstakingaddress` / `createcoldstakingaddress`	New cold-stake delegation address
`fundquantumcoldstakeaddress` / `withdrawquantumcoldstakeaddress`	Fund / unbond-withdraw a delegation
`getquantumcoldstakebalance` / `listquantumcoldstakingdelegations`	Inspect cold-stake holdings
`importquantumcoldstakingdelegation`	Import delegation metadata
`fundquantumoperatorbond` / `withdrawquantumoperatorbond` / `getquantumoperatorbondinfo`	Operate a 30-day operator bond
`getwalletquantumpoolinfo`	Verified operator registry (value, share bps, claims, over-cap)
`getquantumredelegationinfo` / `redelegatequantumcoldstake`	Inspect / execute redelegation (dry-run supported)

9.5 Demurrage

RPC	Purpose
`senddemurrageattestation`	Send a liveness attestation for an eligible wallet-backed direct/tiered v16 key; cold-stake outputs cannot be attested
`getdemurragewalletinfo`	Per-output decay state, effective value, attestation-due flag
`sweepdemurragedecay`	Realize decay on still-spendable outputs and move the effective remainder

9.6 EUTXO inspection and RGB

RPC	Purpose
`createeutxospend` / `createeutxotransition`	Disabled in v30.1.1; always return the v15 ownership-authorization error
`decodeeutxospend` / `verifyeutxospend`	Inspect candidate or known v15-shaped records; never authorize or enable a spend
`importeutxostate` / `listeutxostates`	Persist / list inspection-only EUTXO metadata; does not make an output spendable
`creatergbtransfer` / `acceptrgbconsignment` / `exportrgbconsignment`	RGB transfer lifecycle
`importrgbcontract` / `importrgbassignment` / `listrgbassets`	RGB asset management
`decodergbcommitment` / `verifyrgbconsignment`	Raw RGB inspection

9.7 Wallet maintenance

RPC	Purpose
`deladdressbook`	Remove a sending-address book entry (CLI parity with the GUI)
`burn` / `burnwallet`	Provably destroy specific coins / the whole wallet balance

9.8 The GUI

The Qt wallet adds two dedicated pages:

- **Staking & Mining**, one place for PoS staking, Gold Rush status, the in-process PoW miner, quantum migration, tiered/cold staking, operator bonds, demurrage, RGB, and inspection-only EUTXO metadata. Expensive detail panels load on demand behind a **Refresh details** button so the tab opens instantly even on very large wallets.
- **Account**, a per-family (Legacy / Quantum / Cold-stake / EUTXO) breakdown of every output, its state (bonded / unbonding / withdrawable), and demurrage exposure, with CSV export. An EUTXO row is a frozen-output warning and inspection surface, not a funding or spend workflow.

The **Unlock Wallet** dialog offers two explicit, mutually-exclusive modes: **For Legacy Staking Only** (mint PoS blocks, no spending or quantum actions) and **Legacy and Quantum Staking** (full unlock, required for any quantum, Gold Rush, migration, or cold-staking transaction). Automatic demurrage-attestation attempts additionally require staking to be enabled and a safe spendable fee input.

The Staking & Mining dashboard reports a quarantined claim prominently and identifies the preview command available in the Qt debug console. It does not provide a one-click claim-resolution broadcast button. The same warning applies to CLI and headless daemon operators: signing the guided conflict is a separate explicit action, broadcasting it is another, and neither step guarantees peer acceptance or confirmation.

10. Worked Examples and Community Playbook

Example A, The long-term HODLer (recommended path)

Alice holds 250,000 BLK and wants zero maintenance.

1. **Before V4:** nothing to do. Keep the coins.
2. **During Gold Rush:** she creates and backs up a wallet-backed quantum address. She can dry-run migration planning, but does not fund it yet.
3. **At Migration height 6,193,000 or later:** she runs `migratetoquantum` once. Her coins move to the quantum (v16) address. **She backs up her wallet again** because ML-DSA keys are not in the seed.
4. **Optional:** she runs `fundquantumcoldstakeaddress` to delegate to a cold-staking operator (or her own hot node). Her principal stays owner-controlled and can earn staking rewards. Successful coinstakes refresh the output's activity clock.
5. **Afterward:** if she leaves coins in a direct quantum address, her wallet can attempt an attestation at 3 months while staking is enabled, normally unlocked, and funded by a safe fee input. She still monitors the reported state because an attempt can be deferred. If she delegates, she monitors successful staking or moves the output before prolonged inactivity; delegation alone is not an exemption and cold-stake outputs

cannot be attested.

Alice prepares during Gold Rush, completes one migration action during Migration, and can optionally add a staking workflow.

Example B, The active staker during Gold Rush

Bob holds 40,000 BLK (above the 10,000 whitelist threshold) and runs a node.

1. His account is captured in the whitelist snapshot at height 5,945,000.
2. During Gold Rush he keeps staking. After a qualifying solve, a normally unlocked wallet attempts a QQSIGNAL if that legacy target does not already have an active signal. The confirmed signal remains active for the 14-day window; later solves do not cause one signal per minted block. While active, he shares the PoS pool on top of normal staking rewards.
3. If he wants to also work the PoW lane, he enables `setpowmining`; the light 1-MiB Argon2id puzzle lets his ordinary CPU submit `sendshadowpowclaim` proofs for a share of the PoW pool.
4. He migrates during the 18-month Migration phase, before Final height 6,922,000.

Bob is rewarded precisely for the participation he is already doing.

Example C, The forgotten wallet

Carol migrated to a quantum address during Migration and then lost interest, wallet offline.

- For **six months** after demurrage activates: no effect. 100% retained.
- At **12 months** of total inactivity: 88.9% retained, still barely touched.
- If she returns before the terminal **24-month** boundary and submits a valid liveness attestation, the clock resets before decay is realized. If she spends first, the spend burns the accrued difference and moves only the effective remainder.
- If she reaches a full **24 months** without qualifying activity, the output reaches zero effective value and becomes permanently unspendable. No miner or staker receives it, and Carol is not charged a transaction fee to clean up a zero-value output.

The design gives Carol a long recovery window while permanently removing terminally inactive value from effective supply.

Example D, Running a staking pool

Dan wants to stake on behalf of others.

1. He posts a 30-day operator bond (`fundquantumoperatorbond`), registering a verified commitment.
2. Delegators send him cold-stake delegations; their principals remain theirs. The 20% wallet-policy cap steers new delegations toward under-cap alternatives when they exist, but it does not prevent an operator from exceeding that share.
3. If Dan's node stops producing, a normally unlocked owner wallet can redelegate an eligible, owner-spendable delegation after the clamped zero-win trigger, rate limits, probation, and jitter, but only if a meaningfully better verified target is available. The owner still monitors failures and prolonged inactivity.

11. Economic Analysis: Why This Increases Participation

Every mechanic in V4 pushes the same direction, toward a larger, more active, more distributed set of participants.

- **Gold Rush** front-loads a large, deterministic reward (up to 51,437,700 BLK) at the transition, but pays it *only through staking and mining*. It is the strongest possible incentive to bring nodes online exactly when the network most needs breadth of participation. Holding qualifies; participating pays.
- **Demurrage** removes the free-rider equilibrium of classic PoS. In a passive-holding chain, dormant coins retain full influence forever. Under V4, inactive quantum principal loses effective value and realized decay is burned. Active stakers receive only the ordinary subsidy and explicit transaction fees; their benefit is liveness participation and the deflationary reduction of effective supply, not a transfer from inactive holders.
- **The legacy lockout** guarantees the migration completes, so the network's security actually improves rather than carrying a permanent vulnerable tail. A fully-migrated supply is a stronger, more valuable supply for everyone who stayed.
- **Tiered, cold, and pooled staking** lower the barrier to participation: cold delegation lets a holder participate without exposing the owner key on a hot node, and conditional owner-wallet redelegation can steer stale delegations toward better verified operators.

Across all of these mechanics, the V4 equilibrium points every holder toward the same choice. Whether large or small, technical or not, the sensible move is to participate, because participation is where the rewards are and where the activity clock is refreshed, and where the network's future lies. The "do-nothing" strategy carries the weakest returns, yet it remains trivially easy to leave behind.

In summary, Quantum Quasar rewards HODLers in full for helping secure the network, makes that help nearly effortless, and burns realized decay rather than redistributing it.

12. Security Considerations

- **Quantum resistance is opt-in-by-deadline, not instant.** During Gold Rush and Migration (approximately 720 target days), legacy ECDSA coins remain spendable and therefore quantum-exposed. Holders can prepare addresses and backups during Gold Rush, then should migrate during the 540-day Migration phase. At Final Lockout the spendable path becomes quantum-only. This is an explicit, bounded trade-off in favor of a fair, no-surprises migration.
- **ML-DSA keys are outside the HD seed.** The single most important operational rule: **back up the wallet after every new quantum address.** A seed phrase alone does not recover ML-DSA-protected funds. The wallet enforces "key stored before funds move" and warns at each step, but the backup responsibility is the user's.
- **Attestations and consensus determinism.** Demurrage attestations are ML-DSA-signed and bound to the first input's outpoint as a replay anchor, and are validated in-consensus, so every node computes identical effective values and no attestation can be replayed onto a different coin.
- **Witness v15 is not an enabled smart-contract path.** Its datum/validator commitment lacks ML-DSA owner authorization. v30.1.1 disables supported v15 construction in every phase, and consensus rejects v15 outputs and spends from Migration onward. Its decode, verification, and wallet-metadata surfaces are

inspection-only.

- **A mempool departure is not safe claim abandonment.** A peer can retain a base-valid `QQSPROOF` after the local node removes it. v30.1.1 therefore reserves the exact input until the original or a confirmed conflict resolves it. Guided conflict construction is dry-run by default, requires explicit acknowledgement to sign, never broadcasts, and does not promise a no-loss result.
- **Resource diagnostics are scoped, not consensus permissions.** Optional full-supply scans are single-flight, bounded, progress-reporting, and cooperatively cancellable. One-call operator consent cannot bypass critical record/seek, integrity, storage, snapshot, overflow, shutdown, or cancellation protections. A successful qualification is fixed-height and host-scoped and reports `universal_consensus_bound=false`.
- **Live witness evidence is optional post-release qualification.** When run, the exact-source gate binds the final daemon and CLI to a fresh connected-tip mainnet UTXO MuHash, complete value-bearing witness-v2-through-v16 inventory, and same-tip shadow reconciliation. It requires either no bridge-review outpoints or an approved disposition for every such outpoint. Missing runners, capture paths, maturity, or evidence do not block publication and must not be represented as a successful live qualification.
- **Consensus compatibility is paramount.** Mainnet's whitelist height (5,945,000), Gold Rush boundaries (5,950,000 through 6,192,999), Migration boundaries (6,193,000 through 6,921,999), and Final Lockout height (6,922,000) are consensus rules. The retained timestamp anchors are nominal forecasts, not mainnet phase boundaries. Every node that wishes to remain on the same chain must use identical height values. Operators upgrading or building alternative clients must match them exactly to avoid a chain split.
- **The per-pool cap is policy, not consensus.** It cannot by itself prevent a determined operator from accumulating stake; it only steers the default wallet behavior. Genuine decentralization still depends on delegators choosing diverse operators.

Appendix A: Consensus Constant Reference

Values reflect the v30.1.1 release source. The nominal time anchors and durations descend from v30.1.0. v30.1.1 makes the mainnet lifecycle height-authoritative, starts demurrage automatically at Final Lockout, and adds the competing-claim boundary shown below.

Constant	Value	Meaning	Defined in
`QUANTUM_QUASAR_MAINNET_V4_TIME`	1783835299 (2026-07-12 05:48:19 UTC)	Nominal V4 time anchor; v30.1.1 mainnet lifecycle is height-authoritative	`consensus/params.h`
`QUANTUM_QUASAR_GOLD_RUSH_SECONDS`	15,552,000 (180 days)	Gold Rush duration	`consensus/params.h`
`QUANTUM_QUASAR_MIGRATION_SECONDS`	46,656,000 (540 days)	Migration window	`consensus/params.h`
Nominal final-lockout time	1846043299 (2028-07-01 05:48:19 UTC)	Non-authoritative time-schedule reference (V4 + 720 days)	derived
`SHADOW_WHITELIST_HEIGHT`	5,945,000	Balance snapshot height	`shadow_schedule.cpp`
`SHADOW_WHITELIST_MIN_BALANCE`	10,000 BLK	Whitelist eligibility threshold	`shadow.h`

Constant	Value	Meaning	Defined in
`SHADOW_REWARD_START_HEIGHT`	5,950,000	Gold Rush rewards begin	`shadow_schedule.cpp`
`MAINNET_SHADOW_COMPETING_CLAIMS_ACTIVATION_HEIGHT`	5,993,200	Canonical competing-claim allocation begins	`shadow.h`
`Consensus::Params::nShadowQQP4ActivationHeight`	`INT_MAX` (disabled in v30.1.1)	Separately scheduled exact-input QQP4 activation; not a readiness-bit activation	`consensus/params.h`
`SHADOW_GOLD_RUSH_BLOCKS`	243,000 (180 days)	Gold Rush length	`shadow_schedule.cpp`
`SHADOW_REWARD_END_HEIGHT`	6,192,999	Gold Rush rewards end	`shadow_schedule.cpp`
`MAINNET_QUANTUM_MIGRATION_END_HEIGHT`	6,921,999	Last height-authoritative Migration block	`shadow.h`
`MAINNET_QUANTUM_FINAL_START_HEIGHT`	6,922,000	Height-authoritative Final Lockout and automatic demurrage begin	`shadow.h`
`SHADOW_PHASE1_END_HEIGHT`	6,187,599	Halving phase ends	`shadow_schedule.cpp`
`SHADOW_HALVING_INTERVAL_BLOCKS`	43,200 (~30 days)	Reward-halving period	`shadow_schedule.cpp`
Phase-1 base reward	580 BLK/block, halving	Gold Rush emission (halving)	`shadow.cpp`
Phase-2 base reward	463 BLK/block, flat	Gold Rush emission (tail)	`shadow.cpp`
`SHADOW_MAX_EMISSION`	51,437,700 BLK	Gold Rush issuance cap	`shadow.h`
PoS / PoW split	50% / 50%	Per-block reward pool split	`shadow.cpp`
`SHADOW_SOLVER_ACTIVITY_WINDOW`	18,900 blocks (14 days)	Recent-PoS-solve window for signalling	`shadow.h`
Argon2id (time, mem, lanes)	1, 1024 KiB, 1	PoW puzzle parameters	`shadow.cpp`
`DEMURRAGE_GRACE_BLOCKS`	243,000 (6 months)	No-decay grace period	`consensus/demurrage.h`
`DEMURRAGE_ZERO_BLOCKS`	972,000 (24 months)	Full-decay point	`consensus/demurrage.h`
Decay window	729,000 (18 months)	Quadratic decay span	`consensus/demurrage.cpp`
`DEMURRAGE_ATTEST_VALIDITY_BLOCKS`	243,000 (6 months)	Attestation validity	`consensus/demurrage.h`
`DEMURRAGE_AUTO_ATTEST_BLOCKS`	121,500 (3 months)	Conditional wallet attempt threshold	`consensus/demurrage.h`
`DEMURRAGE_BLOCKS_PER_MONTH`	40,500	Block/month conversion	`consensus/demurrage.h`
ML-DSA public key	1,312 bytes	Quantum public key size	`crypto/mldsa.h`
ML-DSA signature	2,420 bytes	Quantum signature size	`crypto/mldsa.h`
`QUANTUM_MIGRATION_PROGRAM_SIZE`	32 bytes	Direct v16 program	`consensus/quantum_witness.h`
`QUANTUM_TIERED_PROGRAM_SIZE`	40 bytes	Tiered v14/v16 program	`consensus/quantum_witness.h`

Constant	Value	Meaning	Defined in
`EUTXO_PROGRAM_SIZE`	32 bytes	Reserved v15 shape; funding and spending disabled in v30.1.1	`addresstype.h`
`QUANTUM_COLDSTAKE_PROGRAM_SIZE`	32 bytes	Direct v14 program	`consensus/quantum_witness.h`
`QUANTUM_POOL_CAP_BPS`	2000 (20%)	Per-pool delegation cap (policy)	`node/quantum_pool.h`
`QuantumRedelegationPolicy::trigger_multiplier`	6	Expected zero-win interval multiplier	`wallet/redelegation.h`
`QuantumRedelegationPolicy::min_trigger_blocks`	300 blocks	Minimum zero-win trigger	`wallet/redelegation.h`
`QuantumRedelegationPolicy::max_patience_blocks`	4,050 blocks	Maximum zero-win trigger	`wallet/redelegation.h`
`QuantumRedelegationPolicy::rate_limit_blocks`	1,350 blocks	Attempt/success rate limit	`wallet/redelegation.h`
`QuantumRedelegationPolicy::probation_blocks`	1,350 blocks	Current-target activation probation	`wallet/redelegation.h`
`QuantumRedelegationPolicy::stampede_jitter_blocks`	1,350 blocks	Deterministic maximum jitter	`wallet/redelegation.h`
`QuantumRedelegationPolicy::liveness_improvement_blocks`	300 blocks	Required target liveness improvement	`wallet/redelegation.h`
Operator bond period	40,500 blocks (30 days)	Verified operator commitment	`wallet/rpc/staking.cpp`
Block time	64 seconds	Mainnet target spacing	`kernel/chainparams.cpp`
Bech32 HRP	`blk`	Mainnet address prefix	`kernel/chainparams.cpp`

Appendix B: Glossary

- **ML-DSA-44**, Module-Lattice Digital Signature Algorithm (FIPS 204), the post-quantum signature scheme used for all quantum spends and attestations.
- **Gold Rush**, the 180-day bonus-emission epoch that launches V4, paid to stakers and miners.
- **Whitelist**, the deterministic set of scripts holding >=10,000 BLK at height 5,945,000, eligible to earn Gold Rush PoS credits.
- **QQSIGNAL / QQSPROOF**, the OP_RETURN control transactions that claim Gold Rush PoS and PoW rewards respectively.
- **Migration**, moving legacy ECDSA coins into quantum (v16) outputs via `migratetoquantum`.
- **Final Lockout**, mainnet height 6,922,000, after which legacy ECDSA spends are permanently rejected and demurrage begins automatically.
- **Demurrage**, the liveness mechanism by which inactive quantum outputs slowly lose effective value. Realized decay is burned, never paid to a miner or staker. Timely attestation or a spend/recreation refreshes

activity; cold delegation alone does not.

- **Attestation**, a zero-value ML-DSA-signed transaction that resets an eligible direct or tiered v16 key's demurrage clock. Cold-stake outputs cannot be attested. The wallet can attempt attestations only when its staking, unlock, key, fee-input, and capacity prerequisites are satisfied.
- **Cold staking**, owner/staker key separation (witness v14) letting a holder delegate staking while retaining principal control; subject to the inactivity schedule.
- **Tiered staking**, self-staking with a consensus-visible bonding/unbonding lock schedule.
- **Operator bond**, a 30-day verified commitment posted by a staking-pool operator.
- **EUTXO**, a reserved witness-v15 datum/validator commitment shape. v30.1.1 freezes its funding and spending pending a quantum-authenticated ownership design.
- **RGB**, client-side-validated, fixed-supply asset commitments anchored on-chain.

This document describes the Blackcoin Quantum Quasar (Protocol V4) v30.1.1 release. All consensus boundaries are drawn from the v30.1.1 source. Blackcoin is free/open-source software under the MIT license.