BLACK CAT

Proof-of-Work Distribution on Solana

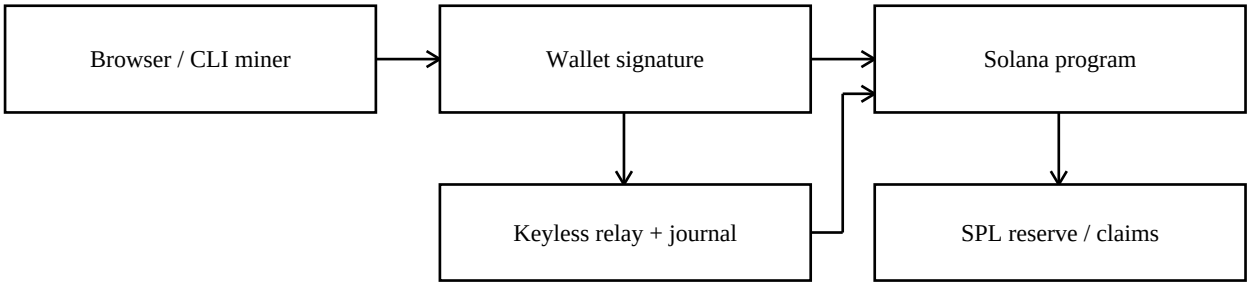
V6.2 Devnet technical specification - 31 October 2026

Abstract

BLACK CAT distributes an existing SPL-token reserve through competitive application-level proof of work. Miners search for a SHA-256 digest at or below a public target. A Solana program verifies the proof, serializes competing winners, records bounded daily reward credits, and later transfers claimed credits from the reserve. It does not create a new consensus blockchain. Solana supplies transaction ordering and finality; the additional work governs token distribution.

The release combines a fixed Nine Lives calendar, bounded future difficulty pins, a later-ancestor challenge, explicit day closure, and expiring claims. Browser and command-line miners use the same proof format. This paper specifies the implemented Devnet design and its limits. It does not claim that the complete reward lifecycle, independent audits or mainnet qualification have been achieved.

1. Architecture and scope



The wallet signs an exact transaction. A client may submit through its RPC transport or use the website's keyless relay. The relay persists the original signed packet before attempting broadcast; it does not hold the miner's key. The program checks accounts, signer, policy, day, current challenge, target and available credit capacity. A locally valid hash alone is neither an accepted win nor a token payment.

The browser implementation and current CLI perform CPU hashing. A GPU implementation could search the same nonce space, but no production GPU miner is claimed here. Holding-history verification and clicking-based work are not prerequisites of this PoW mode. The separate practice miner is nonrewarding and uses a different program and mint.

Specification baseline: source commit 42707bc4dc5690ce8b6f8efb6351a164f46c389e. [1-9]

2. Proof encoding and verification

The program hashes exactly 280 bytes once with SHA-256. Public keys and hashes are their 32-byte binary values, not their textual encodings. The domain includes trailing zero bytes to fill 16 bytes. Round ID is an unsigned little-endian u64; target and digest comparison use unsigned big-endian order. [1]

Offset	Bytes	Field
0	16	BCAT_V62_POW_V1, zero-padded to 16 bytes
16	32	Network genesis hash
48	32	Program ID
80	32	Mint
112	32	Policy hash
144	8	Round ID (little-endian)
152	32	Challenge
184	32	Target (big-endian)
216	32	Miner public key
248	32	Nonce

```
H = SHA256(domain || network || program || mint || policy ||
round_le64 || challenge || target || miner || nonce)

valid mathematical proof iff 0 < T and uint256_be(H) <= T
```

Under the standard uniform-hash model, one independent trial succeeds with probability p = (T + 1) / 2^256. Expected trials are 1/p. After n trials the probability of at least one matching digest is 1 - (1 - p)^n. At a local hash rate h, expected search time is 1/(p*h). These are statistical search estimates, not guaranteed win times: competing miners, stale work, RPC delays and transaction ordering affect acceptance.

Binding the digest to the miner prevents copying a nonce into a proof paying another wallet. Binding network, program, mint and policy separates release identities. The current challenge and round must still match when the transaction executes. Solana serializes transactions that mutate the release; the first admissible winner advances state, and a competing proof for the old state may fail and still incur a transaction fee.

A miner must authenticate the context, rather than accept a target supplied by a web page alone. The reward client checks the pinned network, program linkage, executable hash, mint and policy before producing work. These checks depend on the configured RPC observations; they are not independent proofs of finality.

3. Supply, calendar and reward accounting

The reviewed Devnet genesis maximum is 21,000,000 BCAT with nine decimals: 19,000,000 in the mining reserve and 2,000,000 in the non-mining allocation. Mining releases tokens from that reserve; winning a proof does not authorize minting additional supply. Nine consecutive Lives each last 365 fixed 24-hour days, for 3,285 days. [3,5,6]

Life	Reserve budget (BCAT)	Miner share	Maximum nominal miner pool
I	200,000	60%	120,000
II	400,000	50%	200,000
III	600,000	40%	240,000
IV	1,000,000	34%	340,000
V	1,500,000	30%	450,000
VI	2,000,000	25%	500,000
VII	3,000,000	22%	660,000
VIII	4,300,000	20%	860,000
IX	6,000,000	18%	1,080,000

The nominal full-schedule miner maximum is 4.45 million BCAT and the scheduled burn minimum is 14.55 million. Integer rounding can reduce miner credit. Missing wins and expired claims increase burns. This is not a halving schedule: the share decreases while later Life budgets and nominal miner pools increase. Difficulty follows the separate daily rule in Section 4.

All authoritative arithmetic uses integer base units. For Life budget B, miner basis points b, and zero-based day index j within the Life, the program computes:

```
D = 24 * ceil(B / 8760)
S = min(D, B - min(B, j*D))
C = floor(S*b / 10000)
q = ceil(C / 144)
credit for a win = min(q, C - credited_so_far)
```

S is the scheduled daily slice, C its miner credit cap, and q the published per-win quantum. There are at most 144 wins per day; the last win may receive only the remaining cap. Life I's ordinary daily slice is 547.945205496 BCAT; its miner cap is 328.767123297 BCAT and quantum 2.283105023 BCAT. These figures describe possible credit, not an observed payment or an hourly wage.

4. Difficulty and fresh challenges

The selected Devnet policy starts at an expected one billion hash attempts per proof. It allows an expected-attempt range of one billion to one trillion. The target is fixed for a mining day. A larger target makes work easier; a smaller target makes it harder. No automatic per-Life difficulty doubling is specified for this PoW release. [2,4,5]

Closing source day d computes a nominal target for day d+2; it does not change day d+1. Let w be accepted wins. Elapsed e is the time from opening until the cap was reached for a capped day, otherwise 86,400 seconds. The first two targets are pinned at initialization. For days that have a future pin:

```
R = floor(T_d * 144 * e / (w * 86400)), when w > 0
R = 2*T_d, when w = 0
L = max(floor(T_d/2), floor(T_(d+1)/2), T_min)
U = min(2*T_d, 2*T_(d+1), T_max)
T_(d+2) = min(max(R,L),U)
```

The intersected bounds limit changes against both the source and adjacent nominal targets. A future pin recorded strictly before its opening is timely. Recording at or after opening marks it missed: that day is reward-unavailable, while its nominal target remains available for sequential catch-up. The last two days have no future target to insert. Neither missing operations nor low participation shifts the calendar.

A winner leaves the next round pending with an execution-slot anchor. A later OPEN transaction authenticates Solana's SlotHashes sysvar, requires retained history at or before the anchor, and selects the earliest retained produced ancestor strictly after it. The challenge hashes a separate domain, release identity, day, round, generation, anchor, selected slot and selected hash. The selected slot must precede the OPEN execution slot.

This introduces later public input, not unbiased randomness. Solana block producers can influence or withhold blocks. If retained history no longer covers the anchor, OPEN refuses rather than choosing an arbitrary later boundary or resetting the anchor. The current round may remain unavailable until a legal later day transition. A timely operator is therefore an availability dependency, even though opening is permissionless.

5. Day closure, claims and conservation

Stage	Trigger and effect
PENDING	A fresh challenge is not yet usable; later-ancestor OPEN is required.
OPEN	Valid proofs may consume the remaining daily credit cap.
CAPPED	Daily credit/win limit reached; no further rewarded wins that day.
SEALED	Day closure at or after scheduled end fixes credits and burns S minus credits.
CLAIMED	Each successful own-wallet claim transfers its credited amount once.
EXPIRED	After the deadline, expiry burns the remaining unclaimed credit.

The day closes through an explicit transaction at or after its scheduled end. The program burns S minus credited amounts from the reserve. A claim requires the sealed day, the entitled wallet's signature and the canonical associated token account for the release mint. The website constructs an idempotent associated-account creation instruction followed by the exact CLAIM instruction. Wallet transaction fees and any needed account rent are paid in Devnet SOL. [3,8]

```
deadline(day) = scheduled_end(day) + 259200 seconds
After close: S = burned_at_close + credited
After expiry: S = burned_at_close + paid + burned_unclaimed
```

The 72-hour deadline is measured from scheduled end, not the actual closure time. Closing late does not extend it. A credit, a signed claim, a submitted claim and a finalized token transfer are different states. Expiry is also an explicit lifecycle transaction: describing a scheduled burn does not establish that the burn has already occurred.

The server journal records the original signed claim before attempting broadcast. Recovery identifies it by release, wallet and day; it reconciles that signature rather than generating a replacement. It can therefore be consulted after browser/device loss. An expired, failed or uncertain journaled packet may require reviewed recovery; automatic replacement is deliberately unavailable.

Financial acceptance requires actual transaction receipts and exact account deltas: credited amount, claimant token balance, reserve balance, mint supply, fees and any rent. A locally mined digest or a successful fixture is insufficient evidence. At this paper's snapshot, the live proof-to-credit-to-claim journey remained unverified.

6. Service architecture and security assumptions

The website provides two separate experiences: zero-reward practice mining and the pinned reward release. Both clients hash locally. The reward relay and claim relay run in the Next.js process and hold no miner signing key. Separate journal directories are distinct storage domains, not independent processes or independent custody. Same-origin checks, body limits, bounded concurrency and request limits constrain the web boundary. [7,8]

The operations candidate runs OPEN ticks under a dedicated unprivileged role. It defaults to unsigned preparation. Its proposed oneshot/timer has finite time, tick, disk and fee bounds, serialization, and CPU/memory limits. Missing receipts, uncertain outcomes, expired ancestry and interrupted captures stop the supervisor for manual review. It does not burn, close days, create days, reset anchors or reopen automatically after uncertainty. Installing this source is separate from authorizing a signer or enabling a timer. [9]

Signed intents use exclusive publication and file/directory synchronization on ext4 before sending. This supports conservative original-only recovery; it is not a claim of protection against a compromised host or a proven power-loss storage stack. Installation must verify the immutable source/runtime/dependency inventory, resolved symlinks, ownership and permissions. RPC endpoints are not claimed to have independent operators.

The deployed program is upgradeable. Its upgrade authority can change on-chain code. The currently pinned OPEN payer is also that authority, so placing its key on the VPS would require explicit custody review; the service template must not imply a separate limited key already exists. Client executable-hash checks identify the reviewed binary but do not remove upgrade authority or prevent a dishonest RPC from presenting an incorrect view.

PoW does not establish one-person participation. Faster hardware, parallel miners and specialized implementations can obtain more trials. Browser CPU work is not GPU resistance. Competition and transaction fees may disadvantage small miners; hard puzzles do not guarantee token value or profitability. Mining here does not secure Solana consensus or reproduce Bitcoin's independent chain, longest-chain selection or monetary issuance mechanism.

Principal remaining operational risks include stale proofs, rate-limited RPCs, outages during the short ancestry window, late lifecycle transactions, compromised infrastructure, unauthorized upgrades, and loss of original recovery records. Public Devnet testing must disclose these limits; mainnet requires a separate identity/release review and external approval.

7. Reviewed identity, status and references

This is a dated Devnet technical specification, not a declaration of public-launch or mainnet readiness. As of the 31 October snapshot, the reward program and reserve were initialized; guide, explorer and claims pages were password-protected; reward/claim sending was disabled; no real reward payout was established. Day-one recovery was approved for a later legal window but had not executed. Consult the live explorer and signed/finalized receipts for subsequent state. [7-9]

Public field	Pinned value
Network	Solana Devnet
Program	9hFrNyoid3CPcrhq3861tWi2SMuz9KhMRX43d87Gtttn
Mint	EUPxu3GneEK5RbGNm1FcofVV59e7iFhFx8xtfkCm64hZ
Reserve	Bkv5G3FcEPv7p3hEDpKmyCCZRN4oJrVp9zUjmLybxtZb
First opening	30 October 2026, 20:00 UTC
Policy hash	b7424d6af8b53573e22fbd53cabbbb5a8c90a2aa4b57da0c70fdc5ba3b6f0ddf

References are the project's primary implementation sources at the baseline commit. They support the described implementation, not an external security-audit opinion. The accompanying build receipt records file hashes and the PDF generator version. Source paths below are relative to the repository root.

[1] Proof format
programs/black_cat_paw_work_v62/src/pow_proof.rs

[2] Challenge selection
programs/black_cat_paw_work_v62/src/pow_challenge.rs

[3] Reward/claim lifecycle
programs/black_cat_paw_work_v62/src/pow_tokens.rs

[4] Target arithmetic
programs/black_cat_paw_work_v62/src/pow_target.rs

[5] Selected policy
programs/black_cat_paw_work_v62/src/pow_devnet_policy.rs

[6] Nine Lives budgets
programs/black_cat_paw_work_v62/src/economics.rs

[7] Pinned reward reader
packages/solana-client/src/v62-pow-reward.ts

[8] Claim construction/recovery
packages/solana-client/src/v62-reward-claim-submission.ts

[9] Operations and readiness
scripts/pow/v62-reward-open-worker.mts; ops/solana-v62-testnet/pow-preview/reward-operations/ (supervisor, units and runbook); docs/V62_PUBLIC_DEVNET_LAUNCH_CHECKLIST_2026_10_31.md

Repository: aypeetv-collab/bcatv6. Baseline: 42707bc4dc5690ce8b6f8efb6351a164f46c389e

Pending acceptance: operations installation and custody/fee scope; public-site content review; real supported-wallet signing; hard proof acceptance; exact credit and claim payment; recovery/expiry and release checks; a separate public-ingress decision. External audit, infrastructure/custody/legal review and mainnet release decisions remain separate.