Veld — A Post-Quantum Proof-of-Work Blockchain

Date: 2026-07-07 **Public testnet genesis:** 2026-07-07 02:20:00 UTC **Mainnet launch target:** 2026-08-30 00:00:00 UTC

> Parameters in this document are verifiable against the source at `include/core/constants.h`, `include/consensus/tiers.h`, and `include/consensus/staking.h`. Where the two disagree, the source is authoritative. The public testnet runs the **production** parameter values described here — the same ruleset that launches as mainnet on 2026-08-30.

---

1. Introduction

Veld is a proof-of-work cryptocurrency built around two constraints:

1. **Post-quantum signatures by default.** Every signature on the chain is ML-DSA-65 (Dilithium-3, NIST FIPS 204). There is no secp256k1, no hybrid layer, and no "migrate later" plan. A quantum computer does not invalidate a Veld UTXO.
2. **Open access and longevity over extraction.** Anyone can reach the mining loop on commodity hardware, and the reward mechanisms pay more to participants who stay than to those who extract and leave.

Veld deliberately excludes the features behind the largest failures in other chains: cross-chain bridges, smart-contract VMs, and opaque pre-mines. It is a payment, staking, and validator chain.

---

2. Design Principles

Two principles drive every other decision in this document, and each is enforced by a protocol rule rather than a promise: **everyone can mine, and staying is rewarded.**

2.1 Open access to mining

Mechanism	How it keeps access open
Memory-hard PoW (VeldHash)	A 1 GB memory-hard dataset with an epoch-rotated seed. The hot path fits a commodity desktop and undercuts the ASIC economics that lock out home miners. A laptop CPU and a rented datacenter GPU work the same dataset.
Single-binary client	One Windows/Linux zip, no install, no admin rights, no pool registration, no KYC. The first-run prompt offers resume, keyfile, paste-key, or create-new.
Flat per-block coinbase	The block winner receives a flat 50% of the reward regardless of tier, stake, or history. A first-time miner earns the same per-block cut as a long-tenured one. Tier multipliers apply only to the vault distribution (§4), never to the block race.
Near-miss lottery floor	Miners who never solve a block still get paid for real work. The co-mining pool (20% of every coinbase) pays out every 100 blocks to miners who submitted verified near-misses (§5).
No privileged keys	No pause switch, no upgrade multisig, no foundation mining pool. Every reward stream flows to public addresses reachable by anyone running the standard binary.

2.2 Rewarding long-term participation

Mechanism	How it rewards staying
Mining tier ladder	Five rolling-window tiers (Bronze → Diamond) multiply a miner's vault-distribution share from 1.10× to 3.00×. Thresholds are rolling (7/14, 25/30, 165/180, 335/365, 1000/1095 active days), so a tier must be *held*, not earned once. Miners who fade out demote (§7).
Lockup tier ladder	Stakers who lock for 14 / 30 / 90 days earn 1.10× / 1.25× / 1.50× on every vault distribution for the length of the lock (§4.2).
3.00× ceiling, Diamond-only	Mining tier × lockup tier is capped at `LOCKUP_MAX_MULTIPLIER = 3.00×`, and the cap is reachable only at Diamond. Even Platinum (1.80×) plus a 90-day lock (1.50×) tops out at 2.70×.
Near-miss carry-with-decay	Near-miss credit halves (0.5×) and carries to the next window rather than zeroing, so consistent miners keep a tail-wind and one-shot participants fade out of the lottery within a few cycles.
No pre-mine, no treasury, no founder cut	Every VELD is minted as a block reward to a participant. There is no entity whose balance grows while others mine.

---

3. Tokenomics

Parameter	Value	Notes
Hard cap	21,000,000 VELD	Overflow-checked at consensus in `AddBlock`.
Annual mining emission	550,000 VELD / yr	Linear; no halving discontinuity. The entire 21,000,000 cap is coinbase-issued — no premine, no separate reserve schedule — reaching the cap in roughly 38 years (21M ÷ 550K), leaving the fee market a full generation to mature before the subsidy ends.
Block reward	3.13926940 VELD	`550,000 / 175,200` blocks per year; the sub-unit remainder is paid on the year-boundary block so the annual total is exact.
Block time	180 s	`TARGET_BLOCK_TIME`. Chosen for a low orphan rate and clean Bitcoin-anchoring cadence.
Max reorg depth	100 blocks	Finality floor.
Coinbase maturity	100 blocks	Mirrors the reorg depth.
Pre-mine / founder allocation	0 VELD	None.

3.1 Coinbase split

Every normal block emits 3.13926940 VELD, split four ways at consensus. A miner who tries to keep more than their 50% has the block rejected by every node:

Recipient	Share	Purpose
Block winner	**50%**	Flat direct reward, identical for every miner.
Co-mining pool	**20%**	Accumulated for 100 blocks, paid by lottery (§5).
Vault	**20%**	Distributed to stakers every 480 blocks — daily (§4).
Endorsement pool	**10%**	Distributed to active validators every 480 blocks — daily (§6).
Transaction fees	*(into the vault)*	All fees flow to the vault on top of its 20% share.

Every 100th block routes 100% of its coinbase to the vault.

---

4. Staking

Staking is a commitment: lock coins for a term, earn a share of the vault.

4.1 Lock mechanics

- OP_RETURN payload: `VELD_STAKE|LOCK|<address>|<amount>|<unlock_height>|<tier>`.
- The TX must be signed by `<address>` — consent is checked at consensus, not just in the mempool.
- Each lock is backed by on-chain UTXOs; per-block balance and headroom guards prevent phantom stakes.
- Minimum stake: **1,000 VELD** (`MIN_STAKE_UNITS`). The public testnet runs this production value.

4.2 Lockup tiers

Tier	Lockup	Multiplier
Base	7 days (3,360 blocks)	1.00×
Short	14 days (6,720 blocks)	1.10×
Medium	30 days (14,400 blocks)	1.25×
Long	90 days (43,200 blocks)	1.50×

Multiple stakes from one address blend by amount-weighted average. Each record locks independently; adding a stake does not reset older records' unlock heights.

4.3 Vault distribution

Every 480 blocks (daily, `VAULT_DISTRIBUTION_INTERVAL`), the protocol:

1. Reads the vault UTXO set.
2. Computes each active staker's weight: `stake × mining_tier × lockup_tier`, capped at 3.00×.
3. Applies the whale cap (50% of the cycle budget per staker, active only with ≥5 participants).
4. Pays each staker `cycle_budget × normalised_share`, where the budget ramps from 2% → 8% of the vault over 90 days from staking activation.

All vault accumulation flows to stakers. Validators are paid from a separate endorsement pool (§6).

4.4 Cooldowns

- A cooldown after each stake blocks unstaking regardless of individual lockup expiry.
- UNLOCK requires the same signing-consent check as LOCK.
- Early unstake is rejected at consensus.

---

5. Co-mining — Near-Miss Lottery

The co-mining pool guarantees active miners get paid even in cycles where they never solve a block.

5.1 Near-misses

Every nonce that hashes below `4× target` but above the actual target is a **near-miss** — verifiable proof of real VeldHash work. Each miner broadcasts at most one on-chain near-miss transaction per 100-block window; the chain replays these to build the per-window credit tally that drives the lottery.

5.2 Lottery selection

Every 100 blocks the network runs a flat-uniform lottery.

Parameter	Value
Window	100 blocks (`COMINE_WINDOW_BLOCKS`)
Eligibility	≥1 verified near-miss in the window **and** ≥ `NMS_MIN_BOND_UNITS` staked under the miner's address
Selection	Uniform — every eligible miner has identical probability, independent of how many near-misses they submitted
Slots	**5** winners below `LOTTERY_FLEET_SIZE_THRESHOLD = 1000` eligible miners, **20** at or above it
Payout	The pool is divided into that many equal slots; each drawn winner takes **one slot** (an equal share). Slots that draw no winner **roll forward** into the next window's pool
Replacement	None — no miner is drawn twice in one flush
Randomness	Deterministic, aggregated over the last 8 block headers — identical on every node and verifiable from public data
Carry-over	The credit tally resets after every flush; unfilled slots and under-subscribed pools roll forward and compound

Because the pool divides into a fixed number of slots (not among "however many won"), a single eligible miner takes one slot and the remaining slots roll forward — the pool grows until enough miners participate to fill the draw.

5.3 Why uniform, not proportional

A proportional payout rewards the largest hashers linearly and leaves small miners with rounding dust. A uniform draw among everyone who did real work gives each an identical shot and caps any single participant's expected take, pushing payouts toward the long tail of independent miners.

5.4 Sybil resistance

Winners are drawn **without replacement** and the pool is split across fixed slots, so splitting one worker into N identities does not multiply expected payout — it only dilutes each identity's already-equal chance, while every identity still costs real VeldHash work to produce its near-miss.

---

6. Validators & Endorsement

Validators register on-chain and endorse each block to strengthen finality.

6.1 Registration

- **System activation.** The validator subsystem unlocks together with staking: it stays inert until at least `VALIDATOR_UNLOCK_STAKED = 10,000 VELD` — one minimum validator bond — is staked network-wide. The first validator-scale stake after the staking era opens therefore activates validation; below that, registrations and endorsements are not accepted and the 10% pool simply accumulates.
- OP_RETURN payload: `VELD_VALIDATOR|REGISTER|<pubkey_hex>` — the full 3,904-hex-char (1,952-byte) ML-DSA-65 public key.
- The TX must be signed by the address derived from that pubkey.
- The registrant posts the minimum validator bond into a sigless custody vault (§6.4): **10,000 VELD** (`MIN_VALIDATOR_STAKE`).

6.2 Endorsement

- For each new block, each registered validator signs `(height || block_hash)` with its ML-DSA-65 key.
- The endorsement is broadcast via the mempool and folded into a block (no fee — endorsements are consensus-useful).
- The registry indexes endorsements by block height.

6.3 Reward distribution

The endorsement pool (the 10% coinbase slice) is flushed every 480 blocks (daily), proportional to each validator's endorsement count over the prior 480 blocks. If no registered validator endorsed in the window, the accumulated pool is swept into the vault.

6.4 Bond custody, slashing & yield escrow

Custody, slashing, and the reporter bounty are a single consensus mechanism, live on the chain.

- **Custodial bond.** At registration the bond is paid into a *sigless* custody vault (`STAKE_VAULT_ADDRESS`) — an address with no derivable key, so the principal cannot be moved by anyone, including the validator. It is collateral while validating and is returned in full on a clean deregister at the next 480-block settlement boundary.
- **Slashing (equivocation).** Signing two different block hashes at the same height is cryptographically provable, and anyone may submit the proof. A verified `VELD_VALIDATOR|SLASH` is **permanent**: the pubkey can never re-register, and the offender's stake-record unlock is pushed out by 14,400 blocks (~30 days).
- **Confiscation split.** The custodial bond is split deterministically (integer ppm): **25% to the reporter** (bounty), **25% to the vault**, **50% returned to the offender**. The permanent ban — not full confiscation — is the primary deterrent. The split is replay-deterministic and settles at the

480-block boundary.

- **Yield escrow.** The bonded principal also earns vault yield at the top lockup tier so collateral is not idle. That yield is escrowed and vests on a rolling **~90-day (43,200-block)** delay: vested yield is released to the validator; if the validator is slashed, the still-unvested tail is clawed back with the bond, while already-vested income is kept. A validator who behaves for years and slips once forfeits only the recent unvested tail around the offence.

6.5 Governance

- **General proposals:** `max(7, ceil(active_validators × 0.51))` — an absolute floor of 7 prevents small-cabal capture.
- **Protocol upgrades:** `max(10, ceil(active_validators × 0.67))` — supermajority, with a timelock before activation.
- Voting window: 6,720 blocks (~14 days).

---

7. Mining Tier Ladder

All five tiers are rolling-window — no lifetime milestones. A window is 480 blocks (~1 day); a day is "active" for an address if it produced at least one block.

Tier	Requirement	Multiplier
Base	Mine any block	1.00×
Bronze	7 of last 14 days active	1.10×
Silver	25 of last 30 days active	1.25×
Gold	165 of last 180 days active	1.50×
Platinum	335 of last 365 days active	1.80×
Diamond	1000 of last 1,095 days active	3.00×

The multiplier applies to the **vault distribution weight only** — it never boosts the per-block coinbase. This keeps the block race flat ("everyone can mine") while channelling the long-term incentive into the vault share.

---

8. Post-Quantum Cryptography

8.1 Algorithm

ML-DSA-65 (Dilithium-3), NIST FIPS 204 — chosen for finalised (not experimental) status, lattice-based security (module-LWE / module-SIS) independent of prime-order-group assumptions, and a balanced size/performance tradeoff.

8.2 On-chain footprint

- 32-byte secret seed (the "private key"), same shape as a secp256k1 key for UX parity.
- 1,952-byte public key.
- Signatures up to 3,309 bytes.
- scriptSig ≈ 5.3 KB per signed input.

8.3 Sighash

Varint length prefixes for every count (inputs, outputs, subscript), cryptographically bound to the referenced scriptPubKey to prevent malleability on the signed hash.

8.4 Keyfile format

- ChaCha20-Poly1305 authenticated encryption.
- PBKDF2-SHA256, 600,000 iterations.
- 32-byte salt + 12-byte nonce per file.
- One passphrase unlocks every protocol key on the host, supplied via `EnvironmentFile` so it never transits a command line.

8.5 Browser signing

The wallet embeds a WASM Dilithium bundle (~39 KB, single-file). Browser generation, signing, and verification are identical to the native C++ path.

---

9. Consensus & Network

9.1 Proof of work — VeldHash

- 1 GB memory-hard dataset, deterministically derivable as a ChaCha20 keystream (so verify-only nodes can recompute words on the fly instead of holding the full gigabyte).
- A program of instructions iterated 16 times per hash.
- Epoch-rotated seed (every 256 blocks), keyed by prev_block_hash.
- Truncated Blake2b-512 finalise.

9.2 Block header

88-byte header with a uint64 timestamp (closes the Y2106 cliff) and uint64 nonce (closes the 2^32 ceiling), and a BIP-141-style tagged merkle (`TaggedHash` on leaves and branches, carry-on-odd) that removes the duplicate-last-txid collision surface.

9.3 Difficulty — LWMA (144-block stepped)

Linearly weighted moving average over a 144-block window (T = 180 s), retargeting every 144 blocks with the window's measured time clamped to between 0.5× and 1.5× of target per step. For the first 390 blocks after genesis the chain uses a 36-block window with a wider clamp, so a brand-new network converges to its real hashrate within hours regardless of launch-day participation.

9.4 Reorg & finality

- `MAX_REORG_DEPTH = 100` — any deeper reorg is rejected outright.
- Coinbase maturity: 100 blocks, enforced at consensus.

9.5 P2P layer

- Port 8333 public; the RPC port is loopback-only.
- Seed discovery is DNS-driven (`seeds.veld.network` plus `node1–4.veld.network`), so the seed set changes by DNS edit with no client rebuild.
- Peer-advertised start height is sanity-bounded; ADDR floods are rate-limited; the orphan pool is capped with FIFO eviction.
- A cross-DB atomic commit journal rolls back automatically on a startup crash.

9.6 Client privacy & integrity

- **Tor-ready.** The client ships first-class Tor support — a fetch-pinned Tor daemon and hidden-service reachability — as a one-flag opt-in. While the network is small, clearnet is the default because block gossip over few Tor circuits measurably raises fork rates; the default flips back to Tor as the peer mesh grows.
- **Signed releases.** Auto-updates are gated on an offline ML-DSA-65 release key whose public half is pinned in the binary; the updater verifies the signature over `SHA256SUMS` before it will install anything.
- **Zero-trust sync.** A new node syncs from genesis by default, verifying every proof-of-work and signature itself. Operator snapshots, when published, are signature-checked against a pinned key, genesis-pinned, and re-verified block-by-block in the background after startup (§14).
- **Optional transaction memos.** A sender may attach an optional memo of up to 80 bytes as a value-0 OP_RETURN output. It is byte-checked against the typed text before signing and is never required to transact.

---

10. Genesis & Launch

- **Public testnet genesis:** 2026-07-07 02:20:00 UTC.
- **Genesis message:** "Veld — Where value is earned."
- **Mainnet launch target:** 2026-08-30 00:00:00 UTC.

- **Public testnet (live now):** launched 2026-07-07 on the exact production ruleset — 1,000 VELD minimum stake, 10,000 VELD validator bond, validator system unlocking together with staking, the full Bitcoin peg, smart payments, and the 550,000 VELD/yr emission schedule. What is being rehearsed is mainnet itself, not a simplified stand-in.

Earlier bringup chains were relaunched from fresh genesis several times to ship consensus fixes; each relaunch carries forward wallet keys (addresses are unchanged) but resets block history. The mainnet genesis at 2026-08-30 is the final, non-resetting chain.

---

11. btcVELD — The Bitcoin Peg

btcVELD is an on-ledger token inside Veld consensus where 1 btcVELD is always backed by 1 BTC held in custody. It exists so Bitcoin liquidity can move at Veld speed, settle against VELD in an on-chain market, and inherit Veld's post-quantum transaction layer — while every unit remains redeemable for real BTC.

11.1 The token ledger

btcVELD lives in a consensus-validated token ledger: mint, transfer, and redeem operations ride ordinary Veld transactions as tagged payloads, and every node validates every operation. Balances are integer satoshis; the ledger state folds into the chain's state digest, so any divergence is a consensus fault, not an accounting dispute.

11.2 Wrap and redeem

A *wrap* deposits BTC to the custody address; once confirmed, the issuer signs a mint of the equal btcVELD amount to the depositor's Veld address. A *redeem* burns btcVELD on Veld with a destination Bitcoin script in the burn memo; after the burn is buried past the finality depth, the payout daemon releases the BTC. The issuer key lives on isolated signing hardware and can only sign what the on-chain rules already permit — consensus, not the operator, is the authority on how much btcVELD may exist.

11.3 Consensus protections

- **Hard caps.** Per-mint and total-custody ceilings are enforced by every node. The amount of BTC the system will ever custody is bounded on-chain; a compromised issuer key cannot mint past them.
- **Redeem rate limit.** Redemptions are consensus rate-limited per rolling window (a bounded fraction of outstanding supply per day), so even a full key compromise cannot drain custody in one burst — the network has time to react.
- **Capacity ladder.** Custody capacity grows only as trailing proof-of-work grows: a step function keyed to a multi-day trailing minimum of network work, so capacity can never be flash-inflated by a short burst of rented hash.

11.4 In-consensus Bitcoin verification

Veld nodes maintain a real Bitcoin header chain *inside consensus*. Anyone may relay Bitcoin headers (a permissionless tagged payload); nodes validate proof-of-work continuity and difficulty retargeting from a compiled checkpoint forward. This gives the chain its own view of Bitcoin: deposit proofs can be verified with light-client Merkle proofs against headers the chain itself validated — a trustless mint path that removes the issuer from deposit verification entirely.

11.5 Bitcoin anchoring

Periodically, the Veld tip hash is written into a Bitcoin transaction. Once that anchor is confirmed and proven back to Veld (a Merkle proof against the in-consensus header chain), fork choice refuses any reorganization that would rewind the anchored height. Deep-rewrite attacks against Veld must therefore rewrite Bitcoin as well; the cost of history revision becomes Bitcoin's cost, not a young chain's.

11.6 Solvency watchtower

An independent watchtower continuously compares on-chain btcVELD supply against custody balance on Bitcoin, over an authenticated channel to the issuer's signer. If solvency cannot be proven — or the watchtower goes silent — issuance halts. Fail closed.

11.7 The VELD/btcVELD market

btcVELD trades against VELD in a constant-product on-chain pool validated by every node (the pool's VELD sits under a covenant only a valid pool operation can spend). While the network is young the pool is deliberately capped at 0.005 BTC of btcVELD at risk, and swaps stay locked until VELD has demonstrated real price discovery; adding liquidity is capped, but *removing* it is never gated — funds are never trapped. The swap fee is depth-responsive, set per swap from the pool's btcVELD reserve: 1.00% while the pool holds up to 0.005 BTC, interpolating down through 0.60% at 0.05 BTC to a 0.30% floor at 0.5 BTC and beyond, quantized to 0.05% steps. The fee accrues entirely to liquidity providers inside the reserves; no protocol cut exists. Time-locked provider multipliers remain specified for the swap-unlock release.

---

12. Smart Payments

Veld adds programmable payments without a virtual machine: a small set of covenant opcodes (template-verify plus absolute and relative timelocks) and a distinct pay-to-script address class. Every node enforces the spend conditions; there is no interpreter beyond the script engine, so the attack surface stays proportional to the feature.

- **Escrow.** Funds lock under a multi-signature covenant (for example buyer + seller, with an arbiter tie-breaker). Release requires the agreed signature combination; no other spend path exists.

- **Atomic swaps.** Hash-locked, time-bounded contracts: reveal the preimage and claim, or refund after the timeout. Either both legs of a cross-chain trade complete or both refund — no custodial middleman. The built-in swap desk settles on these.
- **Payment channels.** Two parties fund a 2-of-2 covenant once, exchange signed balance updates off-chain, and settle in one transaction. Publishing a stale state is punishable by a fraud proof during the contest window.

The wallet re-derives every covenant script locally from the user's view of the agreement before signing anything; if the node's proposal does not byte-match the local derivation, the wallet refuses. Fail closed, per policy.

---

13. Routed Payments

Channels compose into a routed payment network. A payment can traverse multiple channels to reach a party with no direct channel, secured hop-by-hop with hash-locks so no intermediary can steal funds in flight. Veld's routing is post-quantum end to end: hop contracts sign with ML-DSA and the route is wrapped in ML-KEM onion encryption, so each intermediary learns only its predecessor and successor — and none of it is retro-decryptable by a future quantum adversary. Watchtower daemons can monitor channels for stale-state closes while their owner is offline.

---

14. Syncing & Trust

The default path for a new node is full initial block download from genesis: every header's proof-of-work and every transaction signature verified locally, trusting no one.

When an operator-published snapshot is available it may be used for fast bootstrap — under three mandatory checks: the manifest must verify against a release key pinned in the binary; the snapshot must declare the exact genesis this binary was compiled for (a mismatched chain is refused outright); and after startup the node **re-verifies the snapshot's proof-of-work block-by-block in the background**, converging to the same zero-trust footing as a genesis sync. A `--full-ibd` flag skips snapshots entirely.

Two further anchors bound history revision: checkpoints compiled into the client pin deep history against fabricated-chain attacks on fresh nodes, and Bitcoin anchoring (§11.5) makes post-anchor reorganizations as expensive as rewriting Bitcoin.

---

15. What Veld is not

By explicit design, Veld does **not** include:

- **Generic cross-chain bridges** — every general-purpose bridge is a multi-audit attack surface. The one external asset Veld carries, btcVELD (§11), is a single-purpose Bitcoin peg with consensus-enforced caps, in-consensus Bitcoin verification, and a hard-bounded blast radius — the opposite of an open bridge.
- **A smart-contract VM** — a VM multiplies the attack surface and dilutes the simplicity that is the primary security argument.
- **A pre-mine or founder allocation** — fixed supply, emitted only by PoW and protocol distribution.
- **Admin override** — no pause switch, no upgrade multisig, no privileged RPC. Governance is the only parameter lever, and it requires on-chain validator quorum.

---

16. References

- NIST FIPS 204 — Module-Lattice-Based Digital Signature Standard (ML-DSA)
- NIST SP 800-132 — Password-Based Key Derivation
- BIP-340 — tagged-hash construction
- BIP-141 — merkle tree shape reference
- RFC 7693 — Blake2
- RFC 8439 — ChaCha20-Poly1305
- Bitcoin white paper — Nakamoto, 2008 (UTXO / PoW reference)
- RandomX specification — Monero Research Lab (memory-hard PoW reference)
- LWMA difficulty algorithm — Zawy, 2018

Every parameter here is verifiable against the source. If this document contradicts `include/core/constants.h`, the source is authoritative.