# TRAIDAG

## Parallel Proof-of-Work. Certified Finality.

A high-rate Layer-1 architecture that separates open block production from deterministic ledger ordering.

| 25 BPS | 2.5 s | 5–7.5 s | 50 / 50 |
|---|---|---|---|
| launch target | cut cadence | modelled finality | reward split |
| parallel work blocks | certified state steps | block admission to execution | work / stake |

### TRAIDAG in one sentence

Miners produce TenQ proof-of-work blocks concurrently. A stake-weighted quorum certifies which blocks enter the ledger, and every full node executes the same committed queue in deterministic order. The work graph carries proposals; it does not choose the ledger history.

**Parallel production**
Permissionless miners can publish work blocks concurrently without competing for a single selected work chain.

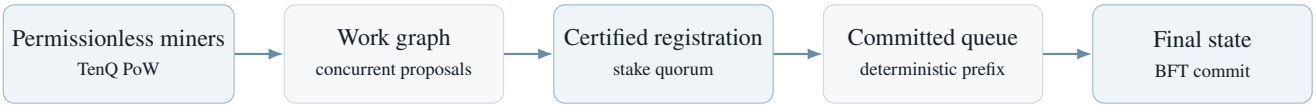
**Deterministic ordering**
Certified registration creates a committed queue whose execution order is identical for every conforming full node.

**Post-quantum profile**
ML-DSA-65 protects transactions and consensus votes, with SHA-384 commitments and hybrid ML-KEM transport.

# 1. Architecture: production and ordering are separate

High-rate proof of work creates a simple tension: the faster blocks are produced, the more often valid work exists concurrently. Traditional designs resolve that concurrency inside the work history itself. TRAIDAG takes a different path. It keeps mining open and parallel, then moves ledger ordering into a stake-certified state machine.

This separation gives each layer one clear job. Miners create valid, work-bound proposals. Validators certify registration and data availability. The committed queue determines what is executed next. Full nodes independently verify the complete transition and resulting state.




Parent links remain useful for relay and reconstruction, but they are not a fork-choice rule and do not determine transaction order.

## A block has a clear lifecycle

| | |
|---|---|
| **Produced** | A miner finds TenQ proof of work for a block bound to a finalised work anchor. |
| **Registered** | A certified cut admits the block into the committed queue. |
| **Executed** | A later cut applies the required queue prefix and commits the resulting state. |
| **Paid** | The block's fixed work claim is materialised once its execution and reward-readiness conditions are satisfied. |

## Why the separation matters

- **Parallel work stays useful.** Concurrent blocks can remain eligible proposals instead of being discarded merely because another block arrived first.
- **Local arrival order is not consensus.** Once the queue is committed, network timing cannot make two correct nodes execute a different order.
- **Finality has a defined event.** Transaction settlement occurs when the execution cut is committed, not after an arbitrary depth of later work blocks.
- **Mining and finality can evolve independently.** Work-block rate and cut cadence are separate protocol parameters, allowing throughput and settlement latency to be tuned on different axes.

**Core distinction.** TRAIDAG is not a work-graph ordering protocol with different parameters. Its defining choice is that the graph carries proof-of-work proposals while the certified queue carries ledger order.

# 2. TenQ: GPU-oriented proof of work

TenQ is the proof-of-work function used by the TRAIDAG work layer. It maps the canonical, anchor-bound work preimage to a deterministic 256-bit value while preserving the familiar nonce, extranonce and target interface used by mining software and pools.

Its design is intentionally broader than a narrow hash pipeline. TenQ combines large reusable epoch memory with nonce-specific writable state, dependent program phases, irregular memory access and mixed integer arithmetic. The objective is **economic specialization resistance**: commodity GPUs should remain efficient, while a specialized implementation must reproduce a balanced mix of memory capacity, bandwidth, on-chip state, arithmetic and control.

## Production profile

| | | | |
|---|---|---|---|
| **Epoch cache** | 32 MiB | **Materialized dataset** | 2 GiB |
| **Dataset node** | 128 bytes | **Scratchpad** | 8 KiB per attempt |
| **Logical lanes** | 32 | **Registers** | 16 × 32-bit per lane |
| **Programs** | 4 dependent phases | **Program length** | 256 instructions |
| **Rounds** | 1,024 per attempt | **Output** | 256-bit work value |

### Full and Light evaluation

A Full evaluator keeps the 2 GiB dataset resident. A Light evaluator stores only the 32 MiB cache and reconstructs requested dataset nodes on demand. Both paths implement the same consensus function and return the same work value for the same input.

### Dependent execution

Later programs are derived from the state produced by earlier phases. This makes the complete instruction schedule attempt-dependent and forces each candidate nonce through the full state transition before its final work value is known.

## Measured performance

| Platform | Backend / path | Batch / workers | Median eval/s | Observed range |
|---|---|---|---|---|
| NVIDIA RTX 2070, Windows | CUDA, 2 GiB profile | 4,096 | 28,964.1 | 28,775.4–29,025.9 |
| NVIDIA RTX 2070, Windows | OpenCL, 2 GiB profile | 4,096 | 28,832.3 | 28,384.8–29,190.6 |
| NVIDIA RTX 2070, Windows | CUDA, 2 GiB profile | 16,384 | 28,260.9 | 28,215.4–28,282.8 |
| NVIDIA RTX 2070, Windows | OpenCL, 2 GiB profile | 16,384 | 28,017.1 | 27,991.7–28,115.7 |
| AMD EPYC 9V74 guest, Linux | Full CPU evaluation | 5 workers | 1,866.6 | 1,739.1–2,006.9 |

The GPU measurements use an NVIDIA GeForce RTX 2070 under Windows. CPU measurements were collected on a constrained EPYC virtual machine. These figures characterize the stated implementations and hardware; they are not difficulty parameters.

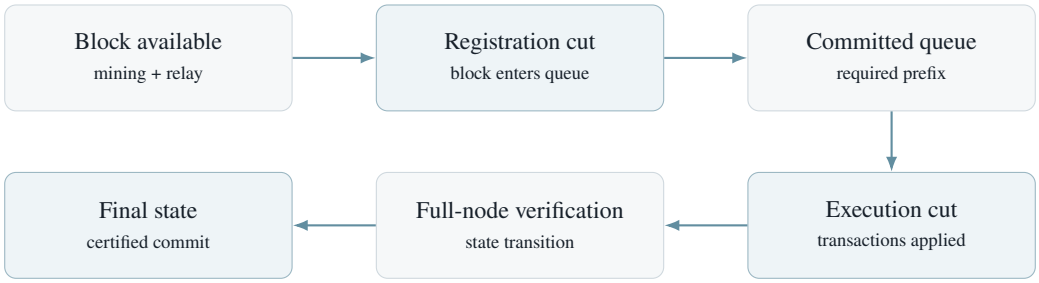
**Why TenQ fits TRAIDAG.** TRAIDAG can admit work at high rates only if block verification remains deterministic and practical across heterogeneous hardware. TenQ keeps the consensus function identical across CPU, CUDA and OpenCL implementations while targeting efficient GPU mining.

For the byte-exact construction, instruction set, Full/Light equivalence, hardware-specialization analysis and quantum-search treatment, see the *TenQ Technical Whitepaper v1.0*.

# 3. Certified ordering, finality and cryptography

TRAIDAG advances ledger state through certified cuts. A cut binds the previous final state, the required execution prefix, newly registered work, the resulting state commitment and the validator set in force. Validators vote on that exact value; every full node then recomputes the transition before accepting it.

## From registered work to final state




The launch profile targets a 2.5-second cut cadence. Under healthy, uncongested conditions, a block available around a proposal boundary crosses roughly two to three cut cadences before its execution commit, giving a modelled **5–7.5 second block-admission-to-finality path**. Mining time before the block exists is separate from this pipeline.

## Consensus security model

**Stake quorum**
A certificate requires more than two thirds of the eligible epoch snapshot by stake weight.

**Deterministic execution**
The predecessor state and committed queue fix the execution set and order; local clocks and local receipt order do not.

**Independent validation**
A certificate authenticates agreement on a cut, but full nodes still verify every transaction, accounting rule and resulting state commitment.

**Stable failure behaviour**
If the required quorum is unavailable, the last committed state remains authoritative until consensus can progress again.

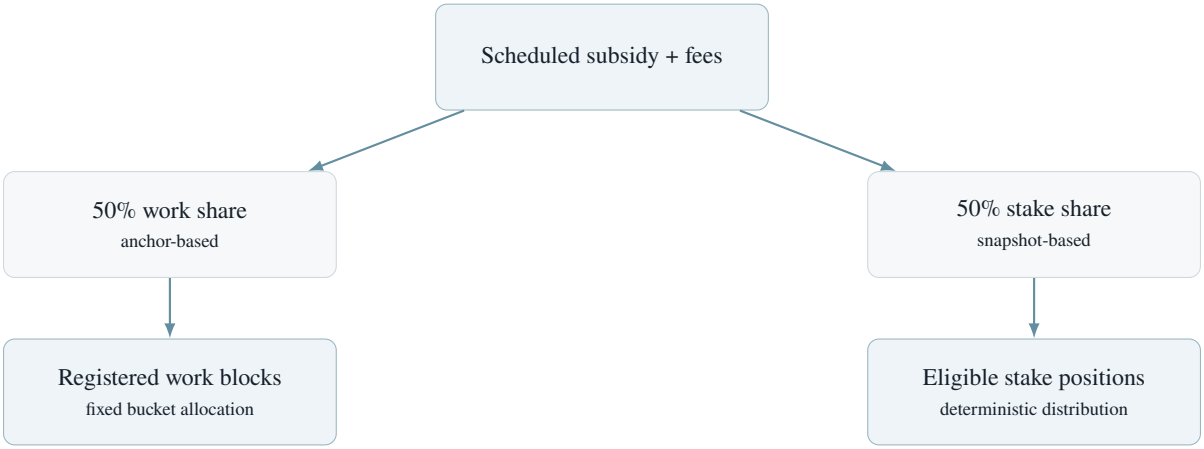
## Post-quantum cryptographic profile

| Purpose | Launch profile |
|---|---|
| Transaction authorisation | ML-DSA-65 |
| Consensus votes and certificates | ML-DSA-65 |
| Consensus commitments | SHA-384 |
| Peer key agreement | TLS 1.3 with hybrid X25519 + ML-KEM-768 |

ML-DSA-65 is deliberately used at the protocol boundary rather than treated as a future migration. The cost is visible and budgeted: signatures and public keys are larger than classical counterparts, so certificate size, relay and storage are part of the launch resource model.

# 4. Mining, staking and reward architecture

TRAIDAG gives proof of work and proof of stake distinct responsibilities and an equal economic share. In the v1 architecture, **50% of scheduled subsidy and 50% of transaction fees go to the work side; the other 50% go to stake**. The split is applied with exact integer accounting so value is neither created nor lost through rounding.




## Anchor-based work rewards

Every work block names an immutable finalised anchor and therefore a fixed target context. Blocks accepted against the same anchor share one scheduled work budget. More parallel work can change how that budget is divided, but it does not multiply aggregate issuance for the anchor.

Reward payment is decoupled from transaction execution. A block can execute before or after its anchor bucket closes; once both conditions are satisfied, the fixed work claim is paid exactly once to the work-bound reward destination.

## Snapshot-based staking

Validator and delegated stake is represented by owned positions recorded in finalised state. Voting power comes from an immutable eligible snapshot for the relevant epoch. Stake rewards are attributed to these historical snapshots and paid through deterministic settlement cursors, preserving the reward weights that were actually in force.

Delegated principal remains owned by the depositor. The reference profile pays rewards directly to owner-controlled reward locks rather than routing all delegated rewards through the validator operator.

## Value remains explicit

Native value is tracked across disjoint committed domains: spendable UTXOs, bonded stake, unbonding claims, work and stake reward reserves, and burned value. Fees move existing value; only the scheduled subsidy path increases minted supply. This keeps mining, staking, payout and slashing accounting within one auditable state transition.

**Network-specific tokenomics.** Asset denomination, genesis allocation, maximum supply, emission schedule, minimum validator bond and initial target are published as network-manifest values. The architecture fixes how those values are enforced; the Litepaper does not substitute generic numbers for them.

# 5. Performance, scaling and the launch profile

TRAIDAG treats work throughput and finality cadence as two separate control axes. A higher work rate increases the number of proposals the network must relay, validate, register and settle. A shorter cut cadence increases the frequency of quorum certificates and reduces the healthy registration-to-execution pipeline. The architecture is designed so either dimension can evolve through a coordinated protocol profile without turning the work graph into a fork-choice mechanism.

## Reference operating profiles

| | Launch profile | Rate-test profile | Cadence-test profile |
|---|---|---|---|
| Work block rate | 25/s | 100/s | 25/s |
| Target cut cadence | 2.5 s | 2.5 s | 1.0 s |
| Registration block cap | 96 | 384 | 96 |
| Execution block cap | 128 | 512 | 128 |
| Registration window | 96 heights | 96 heights | 240 heights |
| Modelled block-to-finality | 5–7.5 s | 5–7.5 s | 2–3 s |

The launch column is the reference v1 operating profile. The other columns illustrate separately scaling work rate and cut cadence; they are protocol test profiles, not automatic runtime modes.

## Designed for worldwide operation

The protocol avoids using local wall-clock time or local block-arrival order as state-ordering inputs. Network distance still matters operationally, so the reference architecture uses bounded objects, announce-then-fetch relay, explicit queue limits, authenticated peer transport and checkpoint/full-history synchronisation modes.

At the ledger level, the important property is that two conforming nodes starting from the same finalised state and validating the same certified cut derive the same next state even if they received the underlying work blocks in a different order.

## What TRAIDAG brings together

- Permissionless, parallel TenQ proof of work
- Stake-certified registration and BFT finality
- Deterministic queue execution
- Post-quantum transaction and vote signatures
- 50/50 work/stake reward architecture
- Anchor-based difficulty and work settlement
- Versioned throughput and cadence profiles
- Full-node validation independent of proposer choice

### The core idea

TRAIDAG does not ask the proof-of-work graph to decide the ledger. Work production stays open and parallel; certified registration creates one queue; deterministic execution turns that queue into one final state. TenQ supplies the GPU-oriented work layer, while stake-weighted consensus supplies the finality boundary.

## Further reading

- *TRAIDAG: Certified Ordering over a Proof-of-Work Block Graph*, Technical Whitepaper v1.2.3, September 2026.
- *TenQ: GPU-Oriented Proof of Work for TRAIDAG*, Technical Whitepaper v1.0, September 2026.