

# Qayyim Whitepaper

---

Version 2.2 — October 2026

The full whitepaper is presented below. [Download the full whitepaper as PDF.](#)

---

## 1. Executive Summary

Current blockchains solve the double-spend problem but fail to address a more fundamental challenge in peer-to-peer commerce: **how do strangers trust each other to transact honestly?** A wallet address reveals nothing about its owner's history or reliability, enabling exit scams, non-delivery fraud, and Sybil attacks via disposable identities.

Qayyim is a blockchain built on **Proof of Trust (PoT)**, a consensus mechanism where validators are selected based on reputation rather than raw capital or computational power. The core premise is that real-world trust — earned through consistent, honest behavior — is the most meaningful resource for securing a decentralized network.

Traditional blockchains face a fundamental trade-off: Proof of Work wastes energy and centralizes around mining pools, while Proof of Stake concentrates power in the hands of capital holders. Qayyim offers a third path where **reputation is the primary qualification for block production**, with economic stake serving as a secondary commitment signal.

### Key Properties

- **Multi-factor trust scoring** — Six weighted factors (transaction quality, counterparty feedback, volume, time decay, vouches, and account age) produce a comprehensive reputation score from 0 to 1000.
- **Sybil resistance by design** — Trust cannot be transferred or traded; it accrues from on-chain history over time. Whether consensus power can be bought is a property of the validator set: it holds once `ValidatorAdmissionV1` is active and admission is open. The chain-1003 testnet starts with five foundation validators, and new ones pass a roster gate, so the property is designed against and not yet demonstrated. Quadratic diminishing returns on vouches make Sybil attacks economically irrational.

- **Immediate finality** — BFT consensus with a two-thirds supermajority ensures blocks are final upon commit, with no forks or reorganizations.
- **Accessible validation** — Designed to run on modest hardware including Raspberry Pi devices, keeping the barrier to entry low.
- **Fixed supply** — 21 million QYM maximum supply with Bitcoin-style halving, creating predictable scarcity.

Qayyim is currently running a live testnet with multiple validator nodes. The codebase is implemented in Rust with a Wasmtime-based smart contract runtime.

---

## 2. Proof of Trust Consensus

### 2.1 Overview

Proof of Trust selects validators based on their accumulated trust scores combined with economic stake. Unlike Proof of Work (which selects by computational power) or Proof of Stake (which selects by capital), PoT introduces reputation as the primary selection criterion.

| Property            | Proof of Work | Proof of Stake   | Proof of Trust                                        |
|---------------------|---------------|------------------|-------------------------------------------------------|
| Primary resource    | Energy        | Capital          | Reputation                                            |
| Barrier to entry    | Hardware      | Tokens           | Time + behavior                                       |
| Centralization risk | Mining pools  | Whale domination | Designed against; unverified (roster-gated set today) |
| Attack cost         | 51% hashrate  | 51% stake        | Trust + stake at slashing risk (§9.4)                 |
| Finality            | Probabilistic | Probabilistic    | Immediate (BFT)                                       |

### 2.2 Validator Selection

Validators are selected for each block using a deterministic, weighted random selection algorithm. The selection weight for each validator is:

```
Selection Weight = trust_score * sqrt(stake)
```

This formula has two critical properties:

1. **Trust is linear** — Doubling your trust score doubles your selection probability.
2. **Stake has diminishing returns** — Doubling your stake only increases selection probability by approximately 41% ( $\sqrt{2} - 1$ ). This prevents capital from dominating reputation.

The selection process works as follows:

1. A deterministic seed is derived from the parent block's validator seed and height using SHA3-256.
2. A ChaCha20 pseudorandom number generator is initialized with this seed.

- Each eligible validator's selection weight is computed as  $\text{trust\_score} * \sqrt{\text{stake}}$ .
- A weighted random selection picks the block proposer.

**Note on formula variants:** The selection weight for block proposer selection uses floating-point arithmetic ( $\text{trust\_score} * \sqrt{\text{stake}}$  in raw base units), which is acceptable because it is evaluated once per block and is not consensus-critical. The BFT voting power formula (Section 2.3) uses integer arithmetic for consensus safety.

This approach provides verifiable randomness — any node can independently verify that the correct validator was selected for a given block.

### Eligibility Requirements

| Parameter              | Value                     |
|------------------------|---------------------------|
| Minimum trust score    | 700 (70% on 0-1000 scale) |
| Minimum stake          | 1,000 QYM                 |
| Maximum validator pool | 100 validators            |

## 2.3 BFT Voting Protocol

Qayyim uses a two-phase BFT voting protocol inspired by Tendermint:

**Phase 1: Pre-Vote** - The selected leader proposes a block. - Validators verify the block and broadcast a signed PreVote. - PreVotes include the block hash (or `nil` if the block is invalid).

**Phase 2: Pre-Commit** - Once a validator sees more than two-thirds of total voting power in PreVotes for a block, it broadcasts a PreCommit. - When more than two-thirds of voting power has PreCommitted, the block is finalized.

**Voting Power** is calculated using the same trust-weighted formula as selection:

$$\text{Voting Power} = (\text{trust\_score} * \text{isqrt}(\text{stake} / \text{ONE\_TOKEN})) / 1000 * 1000$$

This ensures that consensus decisions are weighted by reputation, not just capital.

**Supermajority threshold:** A block requires more than two-thirds ( $>2/3$ ) of total voting power in PreCommit votes to be finalized. This threshold is computed as  $(\text{total\_power} * 2) / 3 + 1$ .

**Equivocation detection:** If a validator signs two different blocks at the same height and round (double signing), this is cryptographically provable and results in slashing.

## 2.4 Block Production

| Parameter                      | Value                |
|--------------------------------|----------------------|
| Block time                     | 12 seconds           |
| Maximum transactions per block | 1,000                |
| Protocol version               | 1                    |
| Finality                       | Immediate (no forks) |

Each block header includes: - Block height and timestamp - Parent hash (chain linkage) - Merkle roots for transactions, state, trust state, dApp state, and receipts - Validator address and trust score at time of production - Next validator seed (for deterministic next-block selection) - BFT commit round and signatures

---

## 3. Trust Score System

### 3.1 Score Calculation

Every account on Qayyim has a trust score ranging from 0 to 1000 (representing 0% to 100%). The score is a weighted combination of six factors:

```
TrustScore = 0.30 * TransactionQuality  
            + 0.20 * CounterpartyFeedback  
            + 0.15 * TransactionVolume  
            + 0.15 * TimeDecay  
            + 0.15 * VouchScore  
            + 0.05 * AccountAge
```

Each factor is independently calculated on a 0-1000 scale before weighting.

## Factor Details

| Factor                | Weight | Scale  | Description                                                                                                                                                                                                                                               |
|-----------------------|--------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Transaction Quality   | 30%    | 0-1000 | Success rate adjusted by dispute and failure penalties. New accounts start at 500 (neutral). 30% penalty per dispute, 50% penalty per failure.                                                                                                            |
| Counterparty Feedback | 20%    | 0-1000 | Time-weighted average of star ratings (1-5) from transaction counterparties. Recent ratings count more than old ones (half-life of approximately 6 months). Accounts with 10 or more lifetime negative ratings are permanently flagged and capped at 250. |
| Transaction Volume    | 15%    | 0-1000 | Logarithmic scaling of successful transactions. 1 tx is approximately 100, 10 tx is approximately 400, 100 tx is approximately 700, 1000+ tx approaches 1000.                                                                                             |
| Time Decay            | 15%    | 0-1000 | Recovery time since last negative event. Exponential recovery: approximately 50% at 30 days, approximately 90% at 90 days, approximately 99% at 150 days. 1000 if no negative events.                                                                     |
| Vouch Score           | 15%    | 0-1000 | Trust-weighted vouches with quadratic diminishing returns (see Section 4).                                                                                                                                                                                |
| Account Age           | 5%     | 0-1000 | Linear scaling up to 365 days, then capped at maximum.                                                                                                                                                                                                    |

## Display Score Curve

Qayyim applies a logarithmic curve to raw scores for display purposes. This makes early tier progression feel more rewarding while keeping premium tiers difficult to achieve. Security-critical thresholds (validator eligibility, vouch permissions) use raw scores to maintain strict requirements.

## 3.2 Trust Tiers

Trust tiers provide a gamified progression system tied to display scores:

| Tier        | Display Score | Description                           |
|-------------|---------------|---------------------------------------|
| Unranked    | 0-499         | New accounts; basic transactions only |
| Bronze      | 500-799       | Established presence                  |
| Silver      | 800-899       | Active, trusted participant           |
| Gold        | 900-939       | Highly trusted                        |
| Diamond     | 940-979       | Elite reputation                      |
| Platinum    | 980-997       | Exceptional track record              |
| Unobtainium | 998-1000      | Near-perfect trust; extremely rare    |

Tier demotions have a 7-day grace period (50,400 blocks at 12-second block times) to prevent temporary score fluctuations from causing abrupt privilege changes.

## 3.3 Role-Based Trust

Trust scores can be filtered by role context (Host, Guest, Worker, General, etc.), allowing counterparties to assess trust for specific interaction types. A user might have high trust as a marketplace seller but lower trust as a service provider. Role-specific feedback is aggregated separately, and the General role serves as a cross-cutting baseline.

## 3.4 Earning and Losing Trust

**Earning trust:** - Complete transactions with positive feedback (star ratings) - Receive vouches from established users - Maintain account age over time - Avoid disputes and failures

**Losing trust:** - Losing disputes in arbitration - Receiving poor star ratings (especially below 3 stars) - Validator misbehavior (slashing events) - Counterparty feedback factors in time-weighted decay, so old negative events gradually lose influence — but accounts flagged as bad actors (10+ lifetime negative ratings) face a permanent cap

---

## 4. Vouching and Accountability

### 4.1 How Vouching Works

The vouching system allows established users to help newcomers build initial reputation by staking tokens as a guarantee of the newcomer's trustworthiness.

When a user vouches for another:

1. The voucher locks stake in escrow (minimum 10,000 base units).
2. If the vouchee misbehaves, the voucher's stake is slashed at 25%.
3. Maximum 10 vouches can be given per user, maximum 5 received.
4. Stake is returned after a 100-block cooldown (~20 minutes) when the vouch is revoked.

### 4.2 Vouching Requirements

| Parameter                 | Value                    |
|---------------------------|--------------------------|
| Minimum voucher trust     | 600/1000 (60%)           |
| Minimum vouch stake       | 10,000 base units        |
| Maximum vouches given     | 10 per user              |
| Maximum vouches received  | 5 per user               |
| Maximum vouch chain depth | 3 hops                   |
| Misbehavior slash rate    | 25% of staked amount     |
| Unlock cooldown           | 100 blocks (~20 minutes) |

### 4.3 Sybil Resistance

The vouch system includes multiple layers of Sybil resistance:

**Quadratic diminishing returns:** Each additional vouch on an account is worth less than the previous one. Vouches are sorted by voucher trust (highest first), and the  $n$ th vouch contributes only  $1/\sqrt{n}$  of the weight of the first:

| Position  | Weight | Cumulative |
|-----------|--------|------------|
| 1st vouch | 100%   | 1.00       |
| 2nd vouch | 71%    | 1.71       |
| 3rd vouch | 58%    | 2.29       |
| 4th vouch | 50%    | 2.79       |
| 5th vouch | 45%    | 3.24       |

Five identical Sybil vouchers provide only approximately 3.24 times the benefit of a single vouch (not 5 times), making Sybil attacks economically inefficient.

**High minimum stake:** At 10,000 base units per vouch, a 5-account Sybil attack costs at least 50,000 base units. A single legitimate vouch from a highly trusted account with higher stake produces comparable trust benefit.

**Trust threshold for vouching:** Vouchers must have at least 60% trust (600/1000), which itself requires genuine participation history.

#### 4.4 Circular Vouch Detection

The system uses depth-limited breadth-first search to detect circular vouching patterns. Before any vouch is created, the algorithm checks whether the vouchee has directly or indirectly vouched for the voucher, up to the maximum chain depth of 3 hops. If a cycle is detected, the vouch is rejected.

#### 4.5 Stake Escrow

All vouch stake is locked in escrow (enforced at the protocol level). Vouchers cannot withdraw their stake while a vouch is active. This prevents withdrawal attacks where a voucher removes their stake after vouching but before the vouchee could misbehave.

---

## 5. Validator Economics

### 5.1 Block Reward Distribution

Each block reward is split among four recipients:

| Recipient           | Share | Era 0 Amount (per block) |
|---------------------|-------|--------------------------|
| Block Validator     | 80%   | 1.6 QYM                  |
| Participation Pool  | 10%   | 0.2 QYM                  |
| Foundation Treasury | 5%    | 0.1 QYM                  |
| Burn                | 5%    | 0.1 QYM                  |

### 5.2 Transaction Fee Distribution

Transaction fees are distributed separately:

| Recipient           | Share |
|---------------------|-------|
| Burned              | 50%   |
| Block Validator     | 40%   |
| Foundation Treasury | 10%   |

Users may also include a priority fee (tip) of up to 1 QYM per transaction, 100% of which goes to the block validator.

### 5.3 Slashing

Validators face escalating penalties for misbehavior:

#### Double Signing (Equivocation)

| Offense | Stake Penalty | Consequence   |
|---------|---------------|---------------|
| 1st     | 10%           | Warning       |
| 2nd     | 25%           | Warning       |
| 3rd+    | 25%           | Permanent ban |

## Downtime (Missed Blocks)

| Offense | Stake Penalty     | Consequence          |
|---------|-------------------|----------------------|
| 1st     | 0% (warning only) | Warning              |
| 2nd     | 5%                | Warning              |
| 3rd+    | 10%               | 30-day temporary ban |

## Invalid Block Production

| Offense | Stake Penalty | Consequence   |
|---------|---------------|---------------|
| 1st     | 15%           | Warning       |
| 2nd     | 30%           | Warning       |
| 3rd+    | 30%           | Permanent ban |

**Additional penalties:** Each slash event incurs a trust score penalty of -50 points. Offense counters reset after a 90-day clean period. Evidence reporters receive 10% of slashed amounts (90% is burned).

## 5.4 Subsidy Schedule

During the first year after mainnet launch, the Foundation provides subsidies to validators:

| Parameter                 | Value               |
|---------------------------|---------------------|
| Duration                  | 1 year from mainnet |
| Subsidy pool              | 1,000,000 QYM       |
| Minimum monthly threshold | 100 QYM             |
| Target monthly reward     | 150 QYM             |

If a validator earns less than 100 QYM per month from block rewards, they receive a subsidy up to 150 QYM total for that month.

## 5.5 Projected APY

| Phase     | Timeline    | Target APY |
|-----------|-------------|------------|
| Bootstrap | Months 0-6  | ~12%       |
| Growth    | Months 6-18 | 6-8%       |
| Mature    | 18+ months  | 3-5%       |

These projections assume a growing validator set and increasing network activity.

---

# 6. Token Economics

## 6.1 Supply Parameters

| Parameter            | Value                        |
|----------------------|------------------------------|
| Token symbol         | QYM                          |
| Decimals             | 18                           |
| Maximum supply       | 21,000,000 QYM               |
| Block time           | 12 seconds                   |
| Initial block reward | 2.0 QYM                      |
| Blocks per year      | ~2,628,000                   |
| Halving period       | ~4 years (10,512,000 blocks) |

## 6.2 Emission Schedule

Block rewards follow a Bitcoin-style halving schedule:

| Era | Years | Block Reward | Approximate Era Emission   |
|-----|-------|--------------|----------------------------|
| 0   | 0-4   | 2.0 QYM      | ~21.0M QYM (capped at 21M) |
| 1   | 4-8   | 1.0 QYM      | ~10.5M QYM                 |
| 2   | 8-12  | 0.5 QYM      | ~5.25M QYM                 |
| 3   | 12-16 | 0.25 QYM     | ~2.6M QYM                  |
| 4   | 16-20 | 0.125 QYM    | ~1.3M QYM                  |
| ... | ...   | ...          | ...                        |
| 10+ | 40+   | 0 QYM        | 0 QYM                      |

The 21 million QYM maximum supply is enforced as a hard cap at the protocol level. The `total_emitted` function ensures cumulative emissions never exceed `MAX_SUPPLY`.

## 6.3 Burn Mechanics

Multiple burn mechanisms create deflationary pressure:

| Source              | Burn Rate                              |
|---------------------|----------------------------------------|
| Block rewards       | 5% of each block reward                |
| Transaction fees    | 50% of all fees                        |
| Slashing            | 90% of slashed stake (10% to reporter) |
| Contract deployment | 50% of 0.1 QYM deploy fee              |

**Projected annual burns (Era 0):** Approximately 263,000 QYM from block rewards alone, plus variable amounts from transaction fees and slashing.

As block rewards decrease through halvings, transaction fee burns become the dominant deflationary force. Long-term circulating supply will stabilize below 21 million QYM due to accumulated burns.

## 6.4 Circulating Supply

$$\text{Circulating Supply} = \text{Total Emitted} - \text{Total Burned} - \text{Locked Stake}$$

Where locked stake includes both validator stake and vouch stake held in escrow.

## 6.5 Foundation Treasury

The Foundation Treasury is funded by: - 5% of block rewards (~263,000 QYM per year in Era 0) - 10% of transaction fees - 50% of contract deployment fees

Treasury funds are allocated to protocol development, security audits, ecosystem grants, marketing, and legal compliance. Governance will transition to community oversight through a planned DAO structure.

## 6.6 Trust-Tiered Fees

Certain operations offer fee discounts based on trust score, rewarding established users:

| Trust Score | App Registration Fee | Discount |
|-------------|----------------------|----------|
| 0-299       | 500 QYM              | 0%       |
| 300-499     | 400 QYM              | 20%      |
| 500-699     | 250 QYM              | 50%      |
| 700-899     | 100 QYM              | 80%      |
| 900+        | 25 QYM               | 95%      |

## 7. Smart Contract Architecture

### 7.1 WASM Runtime

Qayyim runs smart contracts compiled to WebAssembly (WASM) using the Wasmtime runtime engine. Contracts are written in Rust using the Qayyim Contract SDK.

**Key properties:**

- **Deterministic execution** — WASM provides sandboxed, deterministic execution across all validator nodes.
- **Gas metering** — All operations are gas-metered to prevent infinite loops and resource abuse.
- **Memory safety** — Contracts operate in isolated linear memory with configurable limits.
- **Reentrancy protection** — The SDK includes built-in reentrancy guards to prevent common smart contract vulnerabilities.

### 7.2 Contract SDK

The Qayyim Contract SDK provides:

- **`#[contract]` macro** — Defines contract entry points and generates WASM boilerplate.
- **`#[event]` macro** — Structured event emission with indexed topics for efficient querying.
- **Storage API** — Key-value storage with typed serialization.
- **Host functions** — Access to blockchain state including trust scores, balances, block information, and credential verification.
- **Gas API** — Functions for checking remaining gas and estimating costs.

### 7.3 Transaction Types

Qayyim supports a rich set of native transaction types beyond simple transfers:

- **Token operations:** Transfer, stake, unstake
- **Trust operations:** Rate transaction (star ratings), vouch, revoke vouch
- **Smart contracts:** Deploy dApp, call dApp, register off-chain dApp
- **NFT operations:** Mint, transfer, list for sale, buy, offer, bundle operations
- **Name Service (TNS):** Register, transfer, create subdomains, auctions (English and Dutch)
- **Credential system:** Issue, revoke, verify on-chain credentials with governance
- **Zone operations:** Create private zones with configurable membership and rules
- **Profile management:** On-chain profiles with social links and AI consent settings

## 7.4 Transaction Fees and Gas Costs

Each transaction type has a base fee plus an execution fee based on gas consumed:

| Transaction Type | Base Fee   | Gas Cost    |
|------------------|------------|-------------|
| Transfer         | 0.0001 QYM | 21,000 gas  |
| Vouch            | 0.0002 QYM | 30,000 gas  |
| Contract Deploy  | 0.1 QYM    | 100,000 gas |
| Contract Call    | 0.001 QYM  | 50,000 gas  |

Gas is priced at approximately 0.000001 QYM per gas unit (default), with bounds of 0.0000001 QYM (minimum) to 0.0001 QYM (maximum) per gas unit. Additional data costs 16 gas per byte. Users may also include a priority fee (tip) of up to 1 QYM, 100% of which goes to the block validator.

## 7.5 Reference dApps (in development)

Two reference dApps are implemented in-tree as WASM contracts and exercised by the runtime's fixture corpus. **Neither is deployed on the public testnet:** each compiled artifact currently exceeds the per-contract deployment size cap, and reducing them is tracked as TRU-183 / TRU-195.

- **TrustTask** — A trust-aware task marketplace where job completion and payment are mediated by smart contracts with escrow and dispute resolution.
  - **TrustStay** — A short-term accommodation platform demonstrating role-based trust (Host vs. Guest ratings).
-

## 8. Network Architecture

### 8.1 Peer-to-Peer Networking

Qayyim uses **libp2p** for peer-to-peer networking with:

- **Gossipsub** — Publish/subscribe protocol for block and transaction propagation. Validators broadcast new blocks and transactions to the gossip network, ensuring rapid dissemination.
- **Kademlia DHT** — Distributed hash table for peer discovery. New nodes find peers by querying the DHT with their node ID.
- **Protocol versioning** — P2P wire protocol version negotiation (currently v1) allows graceful upgrades. Nodes advertise their protocol version and minimum compatible version during the handshake.

### 8.2 Network Parameters

| Parameter            | Value |
|----------------------|-------|
| Default P2P port     | 9000  |
| Default RPC port     | 8545  |
| Default metrics port | 9615  |
| Testnet chain ID     | 1003  |
| Devnet chain ID      | 31337 |

### 8.3 Bootstrap and Sync

Nodes join the network by connecting to bootstrap peers. The foundation maintains a `bootstrap.json` endpoint that provides current bootstrap peer multiaddresses. Nodes can also specify custom bootstrap peers via the `--bootstrap` command-line flag.

Block synchronization uses a request-response protocol where syncing nodes request block ranges from peers and verify each block's validator signature, BFT commit proof, and state transitions.

### 8.4 Cross-Chain Relay (Planned)

A cross-chain relay system is under development to enable interoperability with other blockchains including Bitcoin and Ethereum. The relay uses SPV (Simplified Payment

Verification) proofs to verify cross-chain transactions without requiring full node operation on partner chains.

---

## 9. Security Model

### 9.1 Sybil Protection

Creating fake accounts is ineffective because:

- New accounts have a trust score of 0 and start Unranked.
- Trust must be earned through genuine transactions and positive feedback over time.
- Vouching requires 60% trust (itself earned through participation) and 10,000 base units of stake.
- Quadratic diminishing returns make multiple low-quality vouches far less effective than single high-quality vouches.
- Bad actor flagging permanently caps counterparty feedback scores after 10 or more negative ratings.

### 9.2 Consensus Security

- **Immediate finality** — BFT with  $>2/3$  supermajority means blocks cannot be reversed once committed.
- **Equivocation detection** — Double signing produces cryptographically verifiable evidence that triggers automatic slashing.
- **Trust + Stake dual requirement** — Attackers must acquire both capital (stake) and reputation (trust score). Building trust score requires months of positive transaction history and vouches from existing trusted users.
- **Deterministic leader selection** — ChaCha20-based PRNG seeded from the previous block ensures verifiable validator selection with no off-chain coordination needed.

### 9.3 Smart Contract Security

- **Reentrancy guards** — The Contract SDK includes built-in reentrancy protection.
- **Gas metering** — All operations consume gas, preventing infinite execution or denial-of-service attacks.
- **Memory isolation** — Each contract runs in its own WASM linear memory sandbox.
- **Signed RPCs** — RPC authentication via API keys prevents unauthorized access to node control endpoints.

## 9.4 Cost of Attack Analysis

**51% Trust Attack:** An attacker would need to control validators holding more than one-third of total voting power. Since voting power depends on both trust and stake, the attacker must: 1. Build trust scores above 700 on multiple accounts (requires months of genuine participation per account). 2. Acquire significant stake for each account. 3. Avoid detection — any slashing event resets the trust-building process.

**Sybil Attack:** With minimum vouch stake of 10,000 units, a 5-account Sybil attack costs at least 50,000 units in locked stake, plus months of trust-building per account. The quadratic diminishing returns on vouches mean 5 Sybil vouches provide only approximately 3.24 times the benefit of one legitimate vouch.

---

## 10. Roadmap

### Current: Testnet

- A multi-validator testnet with BFT consensus and cryptographic block signing — the live node count and the running version are on the [network status page](#), which reads them from the chain rather than restating them here
- Trust scoring, vouching, and tier system operational
- WASM smart contracts, including the TrustTask and TrustStay reference dApps (implemented and fixture-tested in-tree; see §7.5 for their deployment status)
- Mobile wallet on TestFlight
- Foundation website with network status dashboard
- Faucet with proof-of-work protection and IP rate limiting
- NFT system, the name service (TNS), credential framework, and zone system
- Activity simulator for live testnet transactions
- Bootstrap URL support for easy node joining
- Build version negotiation between peers

### Planned: Mainnet Preparation

- Public RPC endpoint with CORS hardening and global rate limiting
- Block explorer for chain browsing
- Security audit of core consensus and smart contract runtime
- Cross-chain relay for Bitcoin and Ethereum interoperability
- DAO governance framework for trust-weighted community voting
- Open-source release and developer documentation portal

### Future

- Mobile wallet public release (App Store and Google Play)
  - Ecosystem grants program funded by Foundation Treasury
  - Protocol upgrade governance via Guardian-tier voting
  - Additional dApp templates and SDK improvements
  - Regional trust network expansion
-

## Appendix A: Protocol Constants

| Constant                  | Value                       | Source                |
|---------------------------|-----------------------------|-----------------------|
| TOKEN_DECIMALS            | 18                          | trustchain-primitives |
| ONE_TOKEN                 | 10 <sup>18</sup> base units | trustchain-primitives |
| MAX_SUPPLY                | 21,000,000 QYM              | trustchain-primitives |
| BLOCK_TIME_MS             | 12,000 (12 seconds)         | trustchain-primitives |
| PROTOCOL_VERSION          | 1                           | trustchain-primitives |
| P2P_PROTOCOL_VERSION      | 1                           | trustchain-primitives |
| INITIAL_BLOCK_REWARD      | 2 QYM                       | trustchain-tokenomics |
| BLOCKS_PER_ERA            | 10,512,000                  | trustchain-tokenomics |
| MAX_ERAS                  | 10                          | trustchain-tokenomics |
| VALIDATOR_THRESHOLD       | 700 (70%)                   | trustchain-primitives |
| VOUCH_THRESHOLD           | 500 (50% display)           | trustchain-primitives |
| DEFAULT_MIN_VOUCH_STAKE   | 10,000 units                | trustchain-trust      |
| DEFAULT_MIN_VOUCHER_TRUST | 600 (60%)                   | trustchain-trust      |

## Appendix B: Trust Factor Weights

| Factor                | Weight | New Account Default       |
|-----------------------|--------|---------------------------|
| Transaction Quality   | 0.30   | 500 (neutral)             |
| Counterparty Feedback | 0.20   | 500 (neutral)             |
| Transaction Volume    | 0.15   | 0                         |
| Time Decay            | 0.15   | 1000 (no negative events) |
| Vouch Score           | 0.15   | 0                         |
| Account Age           | 0.05   | 0                         |

A brand-new account starts with a raw score of **400** ( $0.30 \times 500 + 0.20 \times 500 + 0.15 \times 0 + 0.15 \times 1000 + 0.15 \times 0 + 0.05 \times 0 = 150 + 100 + 0 + 150 + 0 + 0 = 400$ ).

---

*This whitepaper (v2.2) describes the Qayyim protocol as implemented in software version 0.12.1. The protocol is under active development and parameters may change before mainnet launch. All claims in this document have been verified against the codebase.*

*Last updated: March 2026*