COGNITUM™ Whitepaper

Post-Quantum Layer-1 Blockchain

Version 1.6, July 2026

> The post-quantum Layer-1, native DeFi, smart contracts, and on-chain compliance, all without third-party protocols. Every transaction is secured by NIST-standardized cryptography (ML-DSA65 + SPHINCS+) and final in a single block.

> **Project status (July 2026):** COGNITUM is live as a **public testnet** at cognitum.global. An **independent third-party security audit** and the cutover to a **multi-validator, progressively-decentralized mainnet** are planned (§17). Throughout this paper, claims describe the protocol's **design and current testnet implementation**; finality, throughput, and full decentralization are demonstrated at scale only at mainnet. COGNITUM is early-stage infrastructure, extraordinary potential, commensurate risk.

Table of Contents

---

1. Founder

Ryan Paul Gallardo Pillas (Filipino) architected COGNITUM's system end to end, the post-quantum signature model, the consensus and governance design, the fixed-supply integer economy, and the protocol-enforced vesting and treasury controls. Trained as a computer programmer, he completed the 28th Executive Course on National Security (ECNS) in June 2019 at the National Defense College of the Philippines (Camp General Emilio Aguinaldo, Quezon City), and holds a Bachelor of Arts in Religious Education and an honorary Doctorate of Humanities (honoris causa). He is the founding President of One Planet One Earth Foundation Inc. (established June 22, 2020, Philippines), which has held Special Consultative Status with the United Nations Economic and Social Council (UN ECOSOC) since July 2025, a governance-and-mission track record that informs COGNITUM's transparency-first, compliance-native design. The name "COGNITUM", the logo, trademark, and all associated intellectual property are exclusively owned.

---

2. Abstract

COGNITUM is a purpose-built post-quantum Layer-1 blockchain that addresses the most critical unresolved threat in the cryptocurrency industry: the vulnerability of classical cryptographic signatures to quantum computer attacks.

Every major blockchain in existence today, Bitcoin, Ethereum, Solana, BNB Chain, relies on elliptic curve cryptography (ECDSA or EdDSA) for transaction signing. A sufficiently powerful quantum computer running Shor's algorithm can break these signatures, recovering a private key from an exposed public key, allowing an adversary to forge transactions and drain funds from any address whose public key is known. (Block hashing and Proof-of-Work are *not* broken by Shor; they are only modestly weakened by Grover's algorithm and remain secure with larger parameters. The existential threat is to the *signature scheme*, and therefore to account security and the entire asset base of classical chains.)

This is not speculation but settled mathematics, and the resource requirements are falling sharply. Breaking 256-bit ECDSA, the curve securing Bitcoin and Ethereum, is estimated to require on the order of **2,330 logical qubits**, and elliptic-curve cryptography is in fact an *easier* quantum target than RSA. While the largest machines today hold only ~1,000 *noisy* physical qubits, 2025 research cut the physical-qubit requirement roughly **20×** (Google Quantum AI; Gidney, factoring RSA-2048 with under one million noisy qubits), and published fault-tolerance roadmaps, e.g. IBM's ~2,000-logical-qubit system targeted for ~2033, place a cryptographically-relevant quantum computer within the security horizon of long-lived, on-chain assets. Because public keys recorded today can be attacked the moment such hardware arrives, the **"harvest-now, decrypt-later"** threat is a present-tense risk, not a distant one.

COGNITUM is built from the ground up using **ML-DSA65** (CRYSTALS-Dilithium), the signature scheme standardized by the United States National Institute of Standards and Technology (NIST) in FIPS 204 as the primary post-quantum digital signature standard. Every transaction, validator operation, and governance action on COGNITUM is designed to resist both classical and quantum adversaries. No cryptographic system is unconditionally secure; COGNITUM tracks the current NIST post-quantum standards and is built to migrate its algorithms as those standards evolve, it is quantum-*resistant* by design, not a claim of unbreakable or "quantum-proof" security.

Beyond post-quantum security, COGNITUM delivers a complete blockchain ecosystem: native DeFi (staking, lending, swapping, bridging), a governance system, a KYC-integrated compliance layer, and a full COGN-20 token standard, all secured by quantum-resistant cryptography.

Native coin: COGN **Maximum supply:** 1,000,000,000 COGN (1 billion, fixed) **Consensus:** Byzantine Fault Tolerant (BFT) + Block-scoped Proof of History (PoH) **Signature scheme:** ML-DSA65 (NIST FIPS 204)

> *Quantum-threat sources: C. Gidney, "How to factor 2048-bit RSA integers with less than a million noisy qubits" (arXiv:2505.15917, 2025); Google Quantum AI, ~20× reduction in resources to break the Bitcoin ECDLP (2025); IBM fault-tolerance roadmap, ~2,000 logical qubits (Blue Jay, ~2033), large-scale error-corrected system targeted 2028-2029 (MIT Technology Review, 2025). ECDSA-256 ≈ 2,330 logical qubits.*

---

3. The Problem

3.1 The Quantum Threat to Blockchain

In 1994, mathematician Peter Shor published an algorithm that, given a sufficiently powerful quantum computer, can solve the discrete logarithm and integer factorization problems in polynomial time. These are the mathematical foundations of:

- **ECDSA**, used by Bitcoin and Ethereum for transaction signing
- **EdDSA (Ed25519)**, used by Solana, Cardano, and others

- **RSA**, used in TLS and most internet security infrastructure

A quantum computer running Shor's algorithm with on the order of **~2,330 logical qubits** could recover a private key from an exposed public key (breaking 256-bit ECDSA, ECC is in fact an easier quantum target than RSA). Resource estimates are falling sharply, 2025 research cut the physical-qubit requirement roughly 20×, and fault-tolerance roadmaps (e.g. IBM's ~2,000-logical-qubit system targeted for ~2033) place hardware at this scale within the security horizon of long-lived on-chain assets. Google, China's government, and the European Union are investing billions to accelerate this timeline. *(See §2 for sources.)*

The consequence for classical blockchains is catastrophic:

- Any exposed public key (which is exposed every time a transaction is sent) can be used to derive the private key
- An adversary with quantum capability could drain any wallet that has ever transacted
- The total value at risk across Bitcoin and Ethereum alone exceeds $2 trillion

3.2 Why Migration Is Not Trivial

Ethereum developers have discussed quantum migration for years but face fundamental challenges:

- Signature scheme changes require hard forks affecting billions of wallets
- Existing UTXOs and accounts cannot be retroactively protected
- No major chain has completed a post-quantum migration
- Any announcement of "we will migrate" creates a window of vulnerability where adversaries accelerate attacks before migration completes

3.3 The NIST PQC Standardization

In August 2024, NIST finalized its first post-quantum cryptographic standards:

- **FIPS 203**, ML-KEM (Key Encapsulation Mechanism)
- **FIPS 204**, ML-DSA (Digital Signature Algorithm, basis: CRYSTALS-Dilithium)
- **FIPS 205**, SLH-DSA (Stateless Hash-Based Digital Signature)

These are not experimental, they are the official United States government standards for quantum-resistant cryptography, selected for federal and national-security systems. COGNITUM implements FIPS 204 (ML-DSA65) as its primary signature scheme, with FIPS 205 (SLH-DSA / SPHINCS+) available as an optional hash-based alternative (§6.5).

3.4 The landscape at a glance

Every blockchain in production today secures ownership with elliptic-curve signatures, the exact primitive Shor's algorithm breaks. The distinction is not subtle:

Chain	Signature scheme	Quantum-safe signatures	Hashing
Bitcoin	ECDSA (secp256k1)	No	SHA-256
Ethereum	ECDSA (secp256k1)	No	Keccak-256
Solana	Ed25519	No	SHA-256
Cardano	EdDSA (Ed25519)	No	Blake2b
COGNITUM	**ML-DSA-65 + SLH-DSA**	**Yes, NIST FIPS 204/205**	**SHA3-256**

Their *hashing* is already largely quantum-resistant (Grover's algorithm only halves security); it is the **signature layer**, what proves who can move funds, that is exposed. COGNITUM is built on the finalized NIST post-quantum standards at that layer from block zero, with no fallback to elliptic-curve cryptography. As of this writing COGNITUM is among the first Layer-1 networks to do so on the *finalized* (2024) standards; it does not claim to be the first quantum-resistant blockchain (hash-based designs such as QRL predate it).

---

4. COGNITUM Solution

COGNITUM is not a retrofit. It is a greenfield Layer-1 blockchain designed from inception around post-quantum security, with no classical cryptographic debt.

> **Through careful design iteration, we have streamlined the protocol architecture by removing four legacy structural pillars.** This simplification reduces system complexity, lowers operational overhead, and narrows the potential attack surface. Most importantly, this revised foundation allowed us to fully integrate **post-quantum resistant (PQR) cryptography at the core level**. The result is a more efficient, lightweight, and forward-looking network, designed to resist both current cryptographic threats and the long-term risk posed by advances in quantum computing, using NIST-standardized post-quantum cryptography.

4.1 Core Design Principles

Quantum-First: No transaction is ever signed with classical cryptography. ML-DSA65 is the default signature scheme, with SLH-DSA (SPHINCS+, FIPS 205) available as an optional hash-based alternative (§6.5). Every scheme is post-quantum, there is no fallback to ECDSA.

Complete Ecosystem: A blockchain without DeFi, governance, and tooling cannot attract real users. COGNITUM ships with staking, lending, swapping, bridging, governance, KYC, and a full web interface.

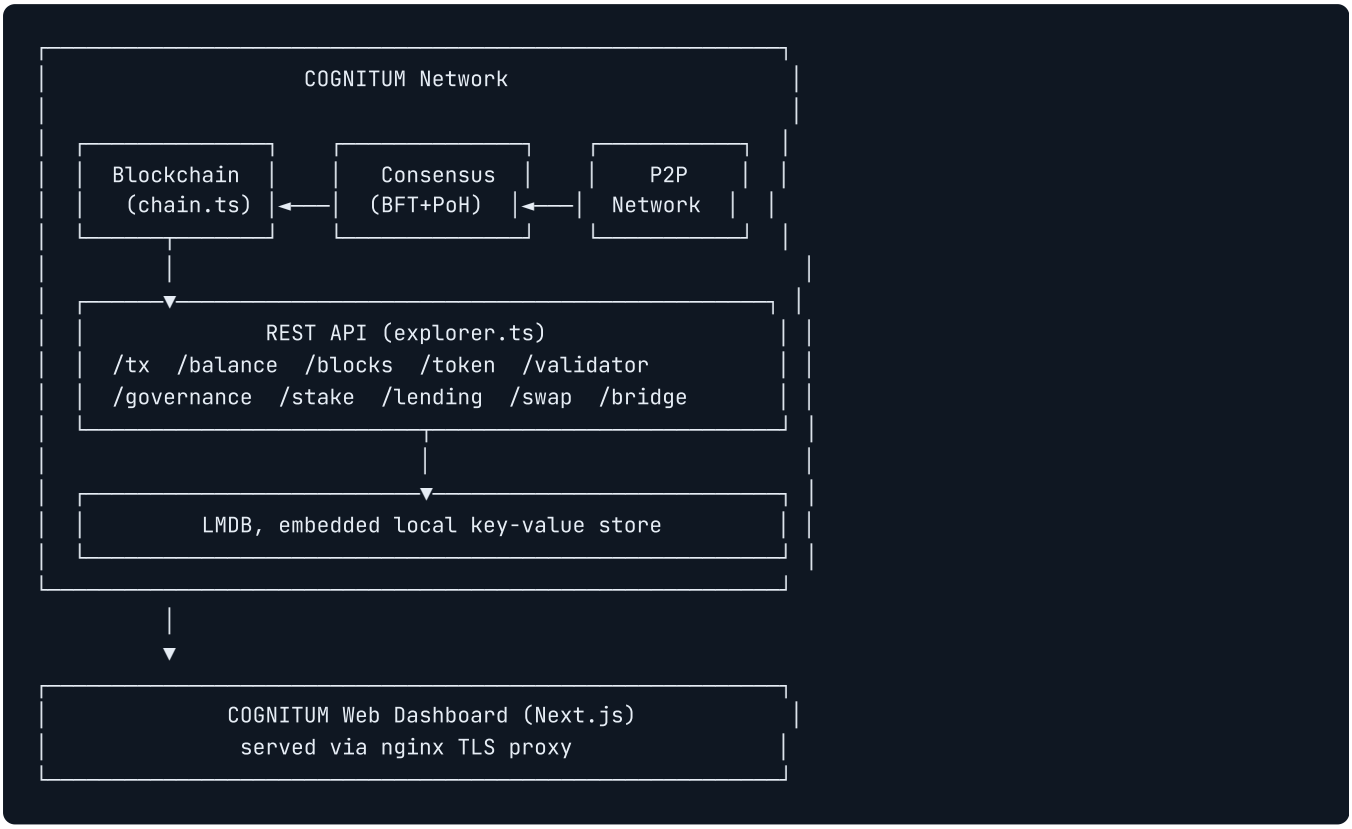
Compliance-Ready: Institutional adoption requires KYC and AML compliance. COGNITUM integrates a webhook-based KYC provider system and a sanctions list checker at the protocol level.

Sustainable Economics: The COGN supply is hard-capped at 1 billion with a Bitcoin-style halving model, ensuring long-term scarcity.

---

5. Architecture

5.1 System Overview



5.2 Technology Stack

Layer	Technology
Blockchain core	TypeScript (Node.js)
Signatures	ML-DSA65 (default) + SLH-DSA / SPHINCS+ (optional, §6.5) via `@noble/post-quantum`
Encryption	ML-KEM768 + AES-256-GCM
Database	LMDB (embedded, per-node key-value store)
API	HTTP/HTTPS REST (Node.js native)
Web UI	Next.js 16 (React, Tailwind CSS, Space Grotesk)
Reverse proxy	nginx (TLS 1.3, hybrid post-quantum key exchange, X25519MLKEM768)
Process management	PM2
Deployment	PM2 on AWS EC2 (Docker Compose for local dev)

5.3 Block Structure

```
interface Block {
  index: number;           // Block height
  timestamp: number;       // Unix milliseconds
  transactions: Transaction[];
  previousHash: string;    // SHA-256 of prior block
  hash: string;            // SHA-256 of this block
  pohHash: string;         // Block-scoped PoH chain root
  miner: string;           // Validator address
  nonce: number;           // Reserved (always 0, COGNITUM uses BFT, not Proof of Work)
  comment?: string;        // Optional block metadata
}
```

5.4 Transaction Structure

```
interface Transaction {
  from: string;       // ML-DSA65 public key (sender address)
  to: string;         // ML-DSA65 public key or username (recipient)
  amount: number;     // COGN or COGN-20 token units
  token?: string;     // Token ticker (omit for COGN)
  fee: number;        // Transaction fee in COGN
  nonce: number;      // Per-address replay protection counter
  publicKey: string;  // Full ML-DSA65 public key
  signature: string;  // ML-DSA65 signature (PQSG-prefixed hex)
  comment?: string;   // Optional memo
  timestamp: number;  // Unix milliseconds
}
```

---

6. Post-Quantum Cryptography

6.1 ML-DSA65 (CRYSTALS-Dilithium)

COGNITUM uses **ML-DSA65**, the medium security level of the Module Lattice-based Digital Signature Algorithm standardized in NIST FIPS 204 (August 2024).

ML-DSA is based on the hardness of the **Module Learning With Errors (MLWE)** and **Module Short Integer Solution (MSIS)** problems over module lattices. These problems are believed to be hard even for quantum computers with access to Shor's and Grover's algorithms.

Key parameters for ML-DSA65:

Parameter	Value
Security level	NIST Level 3 (≈192-bit classical, quantum-safe)
Public key size	1,952 bytes
Private key size	4,032 bytes
Signature size	3,309 bytes
Based on	Module lattice problem (MLWE + MSIS)
NIST standard	FIPS 204

Why ML-DSA65 (not Level 2 or Level 5):

Level 2 offers 128-bit quantum security, sufficient today but conservative given quantum advancement uncertainty. Level 5 offers 256-bit quantum security but with larger keys and signatures. Level 3 (ML-DSA65) provides the optimal balance: 192-bit equivalent quantum security with practical key and signature sizes suitable for a high-throughput blockchain.

6.2 ML-KEM768 (Key Encapsulation)

For encryption (used in secure messaging features), COGNITUM uses **ML-KEM768** standardized in NIST FIPS 203. This is used for:

- Time capsule encryption (encrypt-to-future)
- Secure key exchange in the P2P network layer

6.3 Hashing

COGNITUM uses **SHA3-256** (NIST FIPS 202) as its quantum-safe state and content hash, block and transaction content hashes are produced by a `quantumSafeHash()` helper and tagged with a `QSH-` (Quantum-Safe Hash) prefix for protocol versioning. **SHA-256** (FIPS 180-4) is used internally within the Proof-of-History sequence. Grover's algorithm reduces a 256-bit hash to ≈128-bit effective security, still sufficient for collision resistance. The roadmap includes consolidating all primary hashing on SHA-3 / SHAKE256.

6.4 Signature Workflow

```
1. User generates key pair:
   { publicKey, privateKey } = ML-DSA65.generateKeyPair()

2. Signing message format:
   msg = "{from}|{to}|{amount}|{token}|{fee}|{nonce}"

3. Signature:
   signature = "PQSG-" + ML-DSA65.sign(msg, privateKey).hex()

4. Verification at node:
   valid = ML-DSA65.verify(msg, signature.slice(5), publicKey)
```

The `PQSG-` prefix (Post-Quantum Signature Guard) identifies signatures as post-quantum, enabling future protocol versioning.

6.5 Optional Signature Scheme, SLH-DSA (SPHINCS+)

ML-DSA65 is the default scheme for every account. For participants who prefer **conservative, hash-based security with no lattice assumption**, COGNITUM also supports **SLH-DSA** (SPHINCS+), the stateless hash-based signature standard from NIST FIPS 205. An account opts in at key-generation time; both schemes are first-class on the protocol and verify natively.

COGNITUM uses the **SLH-DSA-SHAKE-192f** parameter set:

Parameter	ML-DSA65 (default)	SLH-DSA-SHAKE-192f (optional)
Family	Module-lattice (MLWE/MSIS)	Hash-based (SPHINCS+)
NIST standard	FIPS 204	FIPS 205
Security level	Level 3 (≈192-bit)	Level 3 (≈192-bit)
Hardness rests on	Lattice problems	SHAKE256 collision/preimage resistance only
Public key size	1,952 bytes	48 bytes
Signature size	3,309 bytes	~35,664 bytes
Signing speed	Fast	Fast (≈0.5 s)

Why offer both. ML-DSA65 and SLH-DSA rest on *different* mathematical foundations. Should an unexpected cryptanalytic advance ever weaken structured lattices, hash-based signatures, whose security reduces only to the collision and preimage resistance of SHAKE256 (which Grover's algorithm merely halves rather than breaking, on current understanding), remain a conservative fallback already deployed and standardized. This **cryptographic diversity** is a hedge: an account securing long-lived, high-value assets can choose the scheme whose assumptions it trusts most.

The trade-off is size. An SLH-DSA signature is ~35.7 KB, roughly **11× larger** than ML-DSA65's already-large signature. Because on-chain signature data is COGNITUM's binding throughput constraint (§16.2), SLH-DSA is intended for **high-value, low-frequency accounts** (treasuries, cold storage, long-term vaults), not high-throughput everyday transactions.

Scheme identification. SLH-DSA signatures carry the `PQSS-` prefix (vs `PQSG-` for ML-DSA65). Nodes auto-dispatch verification on the prefix, so a single transaction format and a single verification path serve both schemes. Encryption (secure messaging) continues to use ML-KEM768 (§6.2) regardless of the signing scheme chosen.

6.6 Hybrid Dual-Signature Accounts (opt-in)

Beyond choosing a single scheme, COGNITUM supports **hybrid dual-signature accounts**, an opt-in account type (the `HQ2-` address prefix) secured by **both** ML-DSA65 (lattice) **and** SLH-DSA (hash-based) at the same time. The address is a SHA3-256 commitment to *both* public keys, so neither key can be substituted, and spending requires *both* signatures to verify over the same message.

An attacker must therefore break **both independent mathematical families at once**: should structured lattices ever be cryptanalytically weakened, the hash-based signature still stands, and the reverse holds as well. No major Layer-1 offers this today.

The trade-offs are the sum of both schemes' signature sizes (about 39 KB) and a 2-of-2 spending requirement (both keys are needed to move funds), so hybrid accounts suit the highest-value treasury, sovereign, and long-lived-asset use cases rather than everyday transactions. The primitive is implemented and property-tested; its integration into the consensus verification path is an audit-gated deliverable that activates alongside the decentralized validator set, and it complements the protocol's crypto-agility so accounts can migrate schemes without a hard fork as NIST standards evolve.

---

7. Consensus Mechanism

COGNITUM uses a two-layer consensus mechanism combining Byzantine Fault Tolerant (BFT) voting with block-scoped Proof of History.

7.1 BFT Validator Voting

Block production follows a **propose-and-vote** model:

```
1. Active validator (proposer) collects transactions from mempool
2. Proposer builds candidate block (up to 500 transactions)
3. Proposer broadcasts vote_request to all active validators
4. Validators verify the block and respond with signed votes
5. Proposer waits for quorum (>⅔ of active validators) within the block interval (~5 s)
6. On quorum: block is finalized and appended to chain
7. On timeout: block is rejected, next validator becomes proposer
```

Quorum rule: A block is finalized when more than ⅔ of active validators cast a `yes` vote, providing Byzantine fault tolerance: with a bonded validator set, the network continues operating correctly as long as fewer than ⅓ of validators are faulty or malicious. The consensus engine implements this in full; its multi-validator fault tolerance becomes active once the independent validator set is bonded at mainnet cutover (§17), and the current testnet runs the same engine with a bootstrap validator set.

Finality: Unlike Bitcoin's probabilistic finality (requiring 6+ confirmations), COGNITUM achieves **single-block finality**, once a block passes BFT quorum it is final and the protocol does not revert it. Nodes apply a fork-choice rule only to *converge* during peer synchronization (e.g. a node rejoining after downtime adopts the canonical chain), never to roll back a quorum-finalized block.

7.2 Fee Market

COGNITUM uses a **dynamic fee market** that scales with mempool congestion:

```
minimum_fee = BASE_FEE × (1 + 9 × (mempool_size / 10,000))     where BASE_FEE = 0.0025 COGN
```

- At 0% mempool utilization: **0.0025 COGN** minimum fee (**≈ $0.0005**, sub-cent, Solana-class, and quantum-safe)
- At 100% mempool utilization: **0.025 COGN** minimum fee (**≈ $0.005**)
- Maximum mempool size: 10,000 pending transactions
- Maximum transactions per block: 500

Fee priority: transactions are ordered by fee (highest first) within each block.

7.3 Block Rewards and Halving

COGNITUM uses a Bitcoin-style geometric halving schedule. Block rewards are issued in **whole COGN** (integer, floored), and the reward halves every **2,000,000 blocks**:

Epoch	Block Range	Block Reward	Issued in epoch
1	0-1,999,999	25 COGN	50,000,000
2	2,000,000-3,999,999	12 COGN	24,000,000
3	4,000,000-5,999,999	6 COGN	12,000,000
4	6,000,000-7,999,999	3 COGN	6,000,000
5	8,000,000-9,999,999	1 COGN	2,000,000
6+	10,000,000 +	0 COGN	0

Supply split. Of the fixed **1,000,000,000 COGN** hard cap, **900,000,000 COGN** is allocated at genesis and **100,000,000 COGN is reserved for block rewards** issued over the schedule above (≈94,000,000 realized, since rewards are floored to whole COGN; the small remainder stays permanently unissued below the cap). Genesis allocation plus block rewards therefore never exceed the 1,000,000,000 COGN ceiling, the protocol caps each reward at the remaining headroom, so issuance can never breach the cap. Beyond the genesis allocation and this block-reward reserve, validator and user staking yield is issued (minted) from a dedicated, bounded **130,000,000 COGN staking-rewards reserve** (§9.5), and referral/welcome bonuses from their own capped buckets; every stream is minted against the same remaining-headroom check, so total issuance across all streams can never exceed the 1,000,000,000 COGN ceiling.

Validator economics. Validators earn the block reward **plus all transaction fees** in the blocks they produce. The reward tail is front-loaded: at the network's block cadence the schedule above is largely exhausted within roughly the first two years, after which **transaction fees become the primary validator incentive**. Sustained fee revenue (not block rewards) is therefore the long-term security budget, and fee parameters are tuned to keep validation economically viable as rewards decline.

---

8. Proof of History

8.1 Block-Scoped PoH

COGNITUM implements **block-scoped Proof of History**, a cryptographic clock that proves the ordering of events within and across blocks.

Each block generates its own PoH chain seeded from the previous block's PoH hash, creating a continuous verifiable timeline across the entire blockchain:

```
Block N-1: pohHash_prev
Block N:   PoH = new ProofOfHistory(pohHash_prev)
           for each tx in block:
               PoH.insertEvent(tx.from + tx.to + tx.amount)
           pohHash_N = PoH.getCurrentHash()
```

Verification: Any node can independently verify that all transactions in a block were processed in the correct order by re-computing the PoH chain from `pohHash_prev` and confirming the result matches `pohHash_N`.

8.2 PoH vs Block Timestamps

Unlike block timestamps (which can be manipulated by validators within a tolerance window), PoH hashes are computationally derived from the actual sequence of events. This makes COGNITUM resistant to timestamp manipulation attacks that affect many classical blockchains.

---

9. Tokenomics

9.1 COGN Overview

Property	Value
Name	COGNITUM Coin
Ticker	COGN
Maximum Supply	1,000,000,000 (1 billion)
Decimals	6 (1 COGN = 1,000,000 integer base units, the ledger stores whole base units for exact, dust-free math, like satoshis/wei; COGN-20 tokens may define 0-18 decimals)
Hard Cap	1,000,000,000 COGN (fixed)
Genesis Allocation	900,000,000 COGN (90%, see §9.2)
Block-Reward Reserve	100,000,000 COGN (10%, issued via halving)
Block Reward (genesis)	25 COGN per block
Halving Interval	Every 2,000,000 blocks
Minimum Transaction Fee	0.0025 COGN ≈ $0.0005 (dynamic, up to 0.025 COGN under load)
Minimum Validator Stake	100 COGN (testnet)

> *Validator stake minimum, testnet vs mainnet:* the 100 COGN minimum applies to the testnet, to keep onboarding open. Before mainnet, the minimum will be raised and **pegged to real value** so the stake is a meaningful security deposit; the minimum is governance-adjustable.

9.2 Supply Distribution

Of the 1,000,000,000 COGN hard cap, **100,000,000 (10%)** is reserved for **block rewards** earned by validators over the halving schedule (§7.3), and **900,000,000 (90%)** is allocated at genesis. A **public token distribution is contemplated following the testnet phase**, structured and conducted only in accordance with applicable securities regulations and subject to qualified legal review in each jurisdiction; the indicative allocation tranches are shown below. **Nothing in this whitepaper is an offer to sell, or a solicitation of an offer to buy, any token or security**, and no price or sale terms are offered herein. The genesis allocation is held and distributed transparently by **COGNITUM** across the buckets below; team and strategic allocations are subject to multi-year vesting to align incentives and prevent supply shocks.

The allocation is deliberately **liquidity-deep and insider-light**: team and strategic backers together hold just **20%**, well below the one-third threshold, and are subject to long, cliffed vesting, while the largest shares go to the **public distribution (10%)**, open-market **liquidity (18%)**, **community & ecosystem programs (11%)**, and **marketing & promotion (7%)**, a combined **46%** public float, to ensure broad distribution from day one. Distribution (percentages of the total 1,000,000,000 COGN supply):

Allocation	% of Supply	COGN	Vesting / Release
Public distribution	10%	100,000,000	Post-testnet, indicative; partial unlock at listing, remainder vested 6-12 months (subject to counsel review)
Liquidity & exchange	18%	180,000,000	At listing
Community & ecosystem	11%	110,000,000	Programmatic over 4-5 years (grants, incentives, airdrops)
Marketing & promotion	7%	70,000,000	Programmatic over 2-3 years (campaigns, listings, growth)
Staking-rewards reserve	13%	130,000,000	Released to supplement validator yield as block rewards decline (§9.5)
Treasury & reserves	11%	110,000,000	COGNITUM-held, governance-controlled, multi-year
Team & core contributors	10%	100,000,000	2-year vest, 1-year cliff
Block rewards (validators)	10%	100,000,000	Halving schedule over network lifetime (§7.3)
Strategic backers	10%	100,000,000	12-month cliff, then 24-month linear vest
Total	**100%**	**1,000,000,000**	

The **Team & core contributors** allocation includes the founder, who holds **7% of total supply (70,000,000 COGN)**, subject to the same 2-year vest / 1-year cliff, with the balance reserved for current and future core contributors. Combined with the founder's ownership of the COGNITUM intellectual property, this keeps founder token holdings deliberately moderate and the cap table insider-light.

9.3 COGN Utility

COGN serves multiple functions within the COGNITUM ecosystem:

- **Transaction fees**, every on-chain transaction requires COGN
- **Validator staking**, validators must bond a minimum of 100 COGN (testnet minimum; the mainnet minimum will be set higher and pegged to real value, and is governance-adjustable)
- **Governance voting**, COGN holders participate in protocol governance
- **Token creation fees**, issuing a COGN-20 token requires COGN payment

- **DeFi collateral**, COGN is the base collateral asset in lending pools
- **Swap pairs**, COGN is the primary pair in the native DEX

Purpose and stewardship. COGN is a **utility asset that secures and governs the network, not an investment product**, its value comes from *using* the network, not from any promise of return. The genesis allocation is held and distributed transparently and on-chain by **COGNITUM**, under multi-year vesting (§9.2), with every treasury movement recorded as an auditable on-chain transaction. Critically, COGNITUM **cannot** mint COGN, breach the 1,000,000,000 hard cap, or alter consensus, these limits are enforced in code, not by trust.

> COGN is used to operate, secure, and govern the network. It is not a security, an investment contract, or a promise of profit, and nothing herein should be construed as such.

9.4 Deflation Mechanisms

- **Fee burning** (roadmap): A percentage of fees can be directed to a burn address
- **Halving**: Block rewards decrease geometrically, reducing new supply over time
- **Hard cap**: No mechanism exists to mint beyond 1,000,000,000 COGN

9.5 Validator Sustainability and the Security Budget

Because block rewards are front-loaded and decline to zero over the halving schedule (§7.3), long-term network security is funded by **transaction fees**, not new issuance. Sustaining validators is therefore a function of usage, the staked value being secured, and the target yield validators require:

$$\text{Required sustained TPS} \approx \frac{r \times S}{(\text{seconds/year}) \times f}$$

where **r** is the target validator APR, **S** is the total staked value, and **f** is the average fee. Infrastructure cost (≈$0.004 of fees per block for 21 validators) is negligible; the binding constraint is delivering a competitive yield on bonded capital. Illustratively, securing $10M of stake at an $0.01 average fee requires ≈2.5 sustained transactions per second, a modest, achievable level; the requirement scales linearly with staked value and inversely with fee size.

To make the transition off block rewards a **glide path rather than a cliff**, COGNITUM dedicates a **130,000,000 COGN staking-rewards reserve** (§9.2). The protocol tops up the shortfall between current block-reward inflows and the genesis-level staking inflow (5 COGN/block) from this reserve: the draw is zero while block rewards are high, rises as they halve, and **self-tapers to zero once the reserve is exhausted**, by which point organic fee revenue is expected to carry the security budget. A dynamic **minimum-fee floor** (0.0025-0.025 COGN) ensures fee revenue cannot collapse to zero. This mechanism is enforced in protocol, not left to discretion.

9.6 On-Chain Transparency: Verifiable Vesting & Multisig Treasury

The allocation and vesting in §9.2 are designed to be **enforced by protocol and auditable by anyone**, *verify, don't trust*, rather than asserted. Three mechanisms make this concrete (rollout at mainnet genesis; see §17):

- **Purpose-tagged allocation accounts.** Each allocation bucket (§9.2) is held in a distinct, publicly readable on-chain account; their balances sum to exactly the 1,000,000,000 COGN cap, so the full distribution can be independently reconciled at any time.
- **Protocol-enforced vesting vaults.** Vested allocations (team, strategic backers, treasury, and program reserves) are locked in on-chain escrow vaults whose releasable amount is computed **deterministically from block time** (cliff, then linear). Because release is enforced in code, **no party, not COGNITUM, not all owners acting together, can release tokens ahead of schedule.** Anyone may trigger a *due* release; no one can accelerate it.
- **M-of-N multisig treasury.** Discretionary treasury movements require a threshold of independent ML-DSA65 signatures (an M-of-N multisig) and are recorded on-chain with a stated purpose, removing single-key control of the treasury.

A public **transparency dashboard** publishes locked-versus-unlocked balances per bucket, upcoming unlocks, and a link to every release and treasury transaction in the block explorer. Combined with the protocol's structural inability to exceed the hard cap or mint at will (§9.3), this converts tokenholder trust from a promise into a continuously verifiable fact.

---

10. COGN-20 Token Standard

COGN-20 is COGNITUM's native token standard, analogous to ERC-20 on Ethereum but secured by post-quantum cryptography and stored natively on the COGNITUM blockchain.

10.1 Token Properties

```
interface TokenDefinition {
  ticker: string;        // 2-8 uppercase letters, unique
  name: string;          // Full token name
  decimals: number;      // 0-18 decimal places
  totalSupply: number;   // Current circulating supply
  initialSupply: number; // Supply at issuance
  issuer: string;        // Issuer's ML-DSA65 public key
  mintable: boolean;     // Can additional supply be minted?
  issuedAt: number;      // Unix timestamp of creation
}
```

10.2 Token Operations

Operation	Description	Authorized By
Issue	Create a new COGN-20 token	Admin account
Transfer	Move tokens between addresses	Any token holder
Mint	Create additional supply	Token issuer (if mintable)
Balance	Query token balance	Public

10.3 Token Fees

Creating a COGN-20 token requires payment in COGN. This creates a revenue stream for the COGNITUM network and prevents spam token creation. Fee schedule is governance-controlled.

---

11. DeFi Ecosystem

COGNITUM ships with a native DeFi suite, all operations secured by ML-DSA65 post-quantum signatures.

11.1 Staking

Validators and delegators can stake COGN to earn block rewards:

- Minimum stake: 100 COGN (testnet; the mainnet minimum will be higher, pegged to real value, governance-adjustable)
- Staking rewards distributed proportionally to stake weight
- Slashing applied for validator misbehavior (double-signing, extended downtime)
- Unbonding period protects the network from rapid stake withdrawal

11.2 Lending Protocol

COGNITUM includes a native collateralized lending system:

- Deposit COGN as collateral
- Borrow against collateral (configurable LTV ratio)
- Interest accrues on outstanding loans
- Liquidation triggers when collateral ratio drops below threshold

11.3 Decentralized Exchange (DEX)

The native COGNITUM DEX uses an **Automated Market Maker (AMM)** model:

- Constant product formula: `x × y = k`
- COGN-paired liquidity pools
- Swap any COGN-20 token against COGN

- Liquidity providers earn swap fees

11.4 Bridge

The bridge module enables cross-chain asset movement:

- Wrapped COGN (wCOGN) on EVM chains (Ethereum, BSC)
- Lock-and-mint mechanism: COGN locked on COGNITUM, wCOGN minted on target chain
- Relayer-based architecture with cryptographic proof of lock
- Quantum-resistant signatures on the COGNITUM side

11.5 Smart Contracts

COGNITUM supports user-deployed **smart contracts**, sandboxed programs that run on-chain and persist their state across the network.

- **Execution model:** a contract is JavaScript executed in a hardened sandbox, a dedicated worker isolate with **no filesystem, network, or process access**, memory-capped and wall-clock-bounded. The contract sees `input` (call arguments), `state` (persistent storage), `result` (its return value), plus a deterministic `block` context (timestamp, height, and a random seed the network agrees on), its own `contract` identity (escrow address + COGN balance), and an `effects` array onto which it pushes value-movement intents. Inputs/state/context cross the sandbox boundary as primitive strings parsed inside the isolate, so no host object is ever reachable, closing the classic `constructor`-walk VM-escape vector at its root.
- **Value transfer:** each contract owns an on-chain **escrow address**; users fund it, and the contract pays out by emitting `effects.push({ type: 'transfer', to, amount })`. The host applies these from the contract's escrow via a balance-checked `releaseFromEscrow`, so a contract can only ever move COGN it actually holds, never mint, overspend, or touch another contract's funds.
- **Deterministic time & randomness:** contracts read `block.timestamp` and a `block.seed` (derived from the block/PoH hash) instead of the host clock or `Math.random`, so every node computes the identical result. For high-stakes randomness, a VRF/commit-reveal can layer on top.
- **Persistence:** contract code and accumulated `state` are stored in the chain's embedded LMDB store and reloaded on node start, so contracts and their data survive restarts exactly like blocks and COGN-20 tokens.
- **Gas:** deployment and each call burn a COGN fee. The deploy fee **scales with bytecode size** (storage cost), like gas on Ethereum or rent on Solana, defaulting to 25 COGN base + 10 COGN/KB, capped at 150 COGN (a typical contract costs ~35-75 COGN); each call costs a flat 1 COGN. All fees are configurable and **burned**, making contract usage deflationary.
- **Access control:** a contract may declare an `allowedCallers` list; if set, only those addresses may execute it. Otherwise it is public.
- **API:** `POST /contract/deploy` (code → contract id), `POST /contract/call` (id + input → result), `POST /contract/fund` (deposit COGN into a contract's escrow); `GET /contract/list` and `GET /contract/{id}` expose the public, transparent contract code and state. See the developer reference for examples.

Deploying an application, step by step. From the **Contracts** page (or the API), a user:

1. **Gets a wallet** and **links** it to their account (proving ownership).
2. **Claims COGN** from the faucet for gas.
3. **Writes** a JavaScript contract over `input` / `state` / `result` (with optional `block`, `contract`, and `effects`).
4. **Deploys** it (`POST /contract/deploy`), a size-based COGN gas fee; receives a contract id.
5. **Funds** the contract's escrow (`POST /contract/fund`) if it needs to pay out COGN.
6. **Calls** it (`POST /contract/call`) to run it; state updates on-chain and any `transfer` effects settle from the contract's escrow.

A conventional web/mobile front-end calls the same API, so end users interact with the application without ever handling contract code directly.

> Roadmap: the v1 engine runs JavaScript for accessibility; a future upgrade replaces the runtime with a WASM virtual machine (e.g. wasmtime) for multi-language support and bit-exact determinism.

---

12. Governance

COGNITUM uses an on-chain governance system for protocol upgrades and parameter changes.

12.1 Proposal Lifecycle

```
Draft → Active → Voting → Finalized
```

1. **Proposal creation**: Any account with sufficient COGN can submit a proposal
2. **Review period**: Governance reviewers (approved validators) review proposals
3. **Voting**: All COGN holders can vote `yes`, `no`, or `abstain`
4. **Finalization**: Proposals pass or fail based on vote count and quorum
5. **Execution**: Passed proposals are implemented via admin actions or automatic parameter updates

Safeguards on parameter changes. Proposals that change protocol parameters (block reward, minimum stake, consensus parameters) carry two protections against governance capture: (1) **bounded ranges**, so governance can tune a parameter only within a safe band and can never, for example, set the block reward high enough to dump the remaining supply in a few blocks; and (2) a **timelock**, so a passing parameter-change proposal is queued and executed only after a delay, giving the network a window to react to a hostile but passing vote. Non-parameter (signal) proposals finalize immediately, and a minimum participation floor denominated in COGN weight prevents any proposal from executing on dust-weight turnout.

12.2 Trustless, Verifiable Voting

Every ballot is **cryptographically signed by the voter's ML-DSA65 wallet key**, in the browser, the private key never leaves the voter's device; only the signature is transmitted. Proposals, votes, and finalizations are recorded **on-chain** as governance events, so the proposal list and vote tally are a deterministic function of block history that anyone can independently recompute by replaying the chain.

The authoritative tally **counts a vote only if it carries a valid signature** that verifies against the voter's address. Unsigned or forged votes are dropped during reconstruction. Because vote events can only be emitted by the node, this signature requirement is what makes the tally operator-proof: **not even the node operator can fabricate or alter a counted vote.**

Token-weighted, snapshot-based tallying. Each vote is weighted by the voter's COGN balance at a deterministic **snapshot**, the block in which the proposal was created, computed by replaying chain history. Quorum and tallies are denominated in COGN weight rather than a count of addresses. This makes governance **Sybil-resistant**: splitting a holding across many addresses yields the same total weight as a single address (weight is conserved), and coins moved *after* a proposal opens carry no weight in it. The snapshot is a pure function of the chain, so any observer recomputes the identical result independently.

12.3 AI-Assisted Governance Review

COGNITUM integrates an Anthropic AI model for preliminary governance proposal analysis. The AI reviews proposals for:

- Technical feasibility
- Economic impact
- Security implications
- Alignment with COGNITUM's post-quantum mission

This is advisory only, all final decisions are made by human voters.

12.4 Treasury & Governance

Authority over COGNITUM is deliberately split between **on-chain governance** (the COGN-holder vote above), **COGNITUM** (the operating entity and treasury custodian), and the **protocol itself** as the final backstop. No single party controls the network.

On-chain governance, the protocol. COGN holders decide protocol matters by vote: economic parameters (fee schedule, staking yield), software upgrades, validator-set policy, and how treasury draws are released. Passed proposals execute as parameter updates; quorum and an AI-assisted review (§12.2) guard against capture and unsafe changes.

The treasury. COGNITUM holds and distributes the genesis treasury **transparently and on-chain**, under multi-year vesting (§9.2). It does **not** have unilateral control of the protocol: it cannot mint COGN, breach the 1,000,000,000 hard cap, or alter consensus, these are enforced in code, not by trust.

Checks and progressive decentralization.

- **Transparency**, every treasury movement is an on-chain transaction anyone can audit.
- **Protocol-enforced limits**, the supply cap, halving schedule, and slashing are code, not discretion; no entity (the company, founder, or validator) can override them.
- **Separation of powers**, the founder and COGNITUM hold the intellectual property and steward the treasury, while COGN holders govern the protocol's parameters.
- **Progressive decentralization**, COGNITUM's role is heaviest at launch and is designed to recede as on-chain governance and the validator set mature, moving more decisions to the community over time.

13. Validator Network

13.1 Validator Lifecycle

```
Applied → Reviewed → Approved → Bonding → Active → Exiting → Inactive
                                                      ↓
                                                   Slashed
                                                   Rejected
```

13.2 Requirements

Requirement	Value
Minimum stake	100 COGN (testnet), mainnet minimum will be higher, pegged to value (governance-set)
Hardware minimum	4 vCPU, 8 GB RAM, 100 GB SSD
Network	100 Mbps uptime, <200ms latency to network peers
Uptime SLA	>95% for active status
Geographic distribution	Validators per region capped at 33% of total

13.3 Slashing Conditions

Validators are slashed (stake reduced) for:

- **Double-signing**: Voting for conflicting blocks in the same round
- **Extended downtime**: Failure to produce blocks when selected as proposer
- **Invalid votes**: Submitting votes that fail verification

13.4 Validator Rewards & Recognition

Validators are the backbone of COGNITUM, and the protocol is designed to **reward their effort fairly today and secure their long-term future in COGN**, entirely within the fixed supply, with no issuance beyond the schedules in §9.

Reward streams (all within the fixed 1B cap). A validator's earnings combine three sources already defined in §9: (1) **block rewards**, 25 COGN/block on the halving schedule (§7.3), paid to block proposers; (2) the **self-tapering draw from the 130,000,000 COGN staking-rewards reserve** (§9.5), which sustains a competitive yield as block rewards decline; and (3) **transaction fees** contributing to the security budget (§9.5). Earnings are distributed in proportion to **bonded stake weighted by reputation** (below), so reliable, long-serving operators earn more per unit of stake than a passive minimum. The protocol targets a **sustainable yield of up to ~10% APR on bonded stake**, a function of usage and staked value (§9.5), not an unbounded emission. (The user staking pool follows a matching lock-term ladder, 1.5% at 1 month, 2% at 3 months, 4% at 6 months, 6% at 1 year, 8% at 2 years, and 10% for a three-year lock.)

Validator Points (on-chain reputation). Each epoch, validators accrue points for availability/uptime and blocks signed, BFT votes cast, **cumulative tenure**, bonded stake, serving an under-represented region (which strengthens decentralization), and a clean, no-slash record. Reputation weights the reward distribution so that *contribution*, not stake size alone, determines share.

Securing the validator's future in COGN:

- **Tenure multiplier**, reward weight rises with continuous honest service (illustratively +5% per quarter, capped), so longevity compounds.
- **Loyalty vesting bonus**, validators past a minimum tenure earn a bonus that vests over multiple years, rewarding commitment.
- **Graceful-exit (retirement) bonus**, a one-time COGN recognition for long-serving validators who exit honestly after the unbonding period.
- **Honest-downtime safety margin**, accidental, non-malicious downtime is penalized far more leniently than Byzantine faults; double-signing and equivocation remain subject to full slashing (§13.3).

Recognition tiers (by tenure and reputation) confer status and a verifiable on-chain badge, never a claim of financial return. All parameters above are set and amendable through on-chain governance (§12), and every reward and slash is recorded as an auditable on-chain transaction.

---

14. Security Model

14.1 Threat Model

Threat	Classical Chain Status	COGNITUM Status
Classical ECDSA break	**Critical**, affects all wallets	**Not applicable**, no ECDSA used
Quantum ECDSA break	**Existential**, all funds at risk	**Resistant**, signed with NIST PQC (ML-DSA65), no ECDSA
51% attack	Possible (PoW chains)	Prevented by BFT quorum requirement
Long-range attack	Possible without checkpoints	Prevented by BFT finality
Replay attack	Possible without nonces	Prevented by per-address nonce counter
Double-spend	Possible (probabilistic finality)	Prevented by single-block BFT finality
Timestamp manipulation	Common exploit	Prevented by PoH
Front-running	Possible	Mitigated by fee-priority ordering

14.2 Network Security

- **TLS 1.3 with hybrid post-quantum key exchange** on all public endpoints, `X25519MLKEM768` (classical X25519 combined with NIST ML-KEM-768, FIPS 203). This protects HTTPS sessions against the **"harvest-now, decrypt-later"** quantum threat at the transport layer, not just on-chain; classical clients fall back to X25519, so there is no compatibility loss. Transport security therefore matches the chain's post-quantum guarantee end to end.
- **Enforced post-quantum P2P**: node-to-node sessions are encrypted with **ML-KEM768 + AES-256-GCM** using ephemeral per-connection keys (forward secrecy). The network can **fail closed** (`REQUIRE_PQ_SESSION`), refusing the plaintext fallback so consensus traffic is never exposed to a harvest-now-decrypt-later observer.
- **Verifiable post-quantum manifest**: a public endpoint plus a one-command verifier (`curl -s https://cognitum.global/verify-chain.mjs | node -`) lets anyone independently confirm the supply cap, the hash-chain linkage, real ML-DSA65 signatures, and that there is **zero classical cryptography in the value path**, turning the post-quantum claim from a promise into a check.
- **nginx** reverse proxy with security headers (HSTS, CSP, X-Frame-Options)
- **Rate limiting** on resource-intensive read endpoints and sensitive owner endpoints (per-minute caps), plus per-peer P2P message rate limits
- **JWT authentication** with 2-hour session expiry
- **Account lockout** after 10 failed login attempts (15-minute lockout)
- **bcrypt** password hashing with per-user salt

14.3 Third-Party Audit

COGNITUM is seeking security audits from recognized firms specializing in blockchain and post-quantum cryptography prior to mainnet launch. Audit reports will be published publicly.

14.4 Security Assumptions & Limitations

COGNITUM is engineered to be **quantum-resistant, not "quantum-proof."** We state our assumptions and limits plainly:

- **Cryptography.** Account, validator, and governance signatures use NIST-standardized post-quantum algorithms (ML-DSA65 / FIPS 204, optional SLH-DSA / FIPS 205), and key exchange uses ML-KEM (FIPS 203). Their security rests on the best current cryptanalysis; **no cryptographic system is unconditionally secure.** Should a scheme be weakened, the protocol is designed to migrate algorithms (and offers hash-based SLH-DSA as an independent fallback).
- **Consensus.** Single-block BFT finality, fork-resistance, and double-spend prevention hold under the standard **honest-supermajority assumption**, at least two-thirds of the active validator stake behaving correctly, with the ≤1/3-per-region cap limiting geographic/operator concentration. An adversary controlling more than one-third can stall liveness, and more than two-thirds can violate safety, as in any BFT system.
- **Cross-chain bridges.** Transfers to/from external chains (e.g. Ethereum, Bitcoin) **inherit the security of those chains**, which are *not* post-quantum, and the bridge relayer signs external-chain transactions with that chain's native (classical) cryptography. Assets in transit depend on the counterparty chain and the bridge's liveness; only the COGNITUM side is post-quantum.
- **Smart contracts.** Deployed contracts run as written in a resource-limited sandbox. Logic errors are the deployer's responsibility, the platform isolates contracts but does not guarantee any individual contract is bug-free.
- **Self-custody.** Control of an ML-DSA65 private key is control of the funds. Keys are generated and held by users; a lost key means permanently lost funds, and COGNITUM cannot freeze, reverse, or recover them.
- **Maturity.** COGNITUM currently operates on **testnet**; validator decentralization, economic parameters, and third-party audits are being finalized before mainnet. Testnet COGN carries no monetary value.

---

15. Compliance and KYC

15.1 KYC Integration

COGNITUM includes a native KYC (Know Your Customer) webhook integration:

- Compatible with Jumio, Onfido, Sumsub, and other major KYC providers
- KYC status stored on-chain per address

- Transactions from KYC-rejected addresses can be blocked at protocol level
- HMAC-SHA256 webhook signature verification

15.2 Sanctions Compliance

COGNITUM operates a **sanctions list checker** at the transaction submission layer:

- Addresses appearing on OFAC, UN, and EU sanctions lists are blocked
- Sanctions list updated regularly from operator-supplied data
- Configurable enforcement levels (warn / block)

15.3 Regulatory Positioning

COGNITUM is designed to be **compliance-first**, enabling institutional adoption in regulated environments. The combination of KYC, sanctions checking, and full transaction auditability positions COGNITUM as suitable for:

- Regulated financial institutions
- Government digital asset programs
- Enterprise tokenization platforms

---

16. Layer-2 Scaling Architecture

COGNITUM's Layer 1 (L1) is a settlement and security base, not a ceiling. This section defines the Layer-2 (L2) scaling path: protocols that execute transactions off the main chain and periodically settle a compressed, cryptographically proven summary back to L1. L2 activity inherits L1's post-quantum security guarantees while relieving on-chain congestion and enabling near-zero-fee, high-frequency transactions.

16.1 Design Goals

Goal	Requirement
Security inheritance	L2 derives its safety from L1, no weaker cryptography, no new majority-trust assumption
Quantum safety end to end	No elliptic-curve or other Shor-vulnerable primitive anywhere in the L2 or its proofs
Congestion relief	Move high-frequency activity off L1 while preserving final settlement on L1
Micro-fees	Per-transaction cost approaching zero through batch amortization
Trustless exit	Users can always withdraw to L1, even if the L2 operator misbehaves or halts

16.2 Why a Post-Quantum Chain Needs Layer 2 *More*, Not Less

The defining constraint of COGNITUM is signature size. An ML-DSA65 signature is **3,309 bytes**, roughly **50× larger** than a 64-byte ECDSA signature. A full L1 block of 500 transactions therefore carries on the order of **1.6 MB of signature data alone**.

This means on-chain data becomes the binding throughput constraint for COGNITUM far sooner than for a classical chain. The single biggest win of a rollup, replacing thousands of individual signatures with one succinct validity proof, is consequently **more valuable on COGNITUM than on any ECDSA/EdDSA chain**.

Metric	Classical chain (ECDSA)	COGNITUM L1 (ML-DSA65)	COGNITUM L2 (rollup)
Per-tx signature footprint	~64 bytes	3,309 bytes	amortized to near-zero
10,000 signatures on-chain	~640 KB	~33.1 MB	one validity proof (≈ tens-hundreds of KB)
Throughput ceiling driver	execution	**signature data**	proof size + state diff

16.3 The Quantum-Resistant Layer

The L2 is made quantum-resistant in two independent parts:

1. Signature inheritance. Every L2 transaction is still signed with ML-DSA65 and carries the `PQSG-` prefix, exactly as on L1 (§6.4). Users do not adopt any new key type to transact on L2.

2. Post-quantum proof system. This is the critical design choice. Mainstream succinct proof systems, Groth16, PLONK, KZG commitments, rely on **elliptic-curve pairings**, which Shor's algorithm breaks. Adopting them would silently reintroduce exactly the classical vulnerability COGNITUM was built to eliminate. COGNITUM L2 instead uses **hash-based STARKs** (FRI-based proofs) constructed entirely on SHA3-256 / SHAKE256. Their security rests *only* on hash collision and preimage resistance, which Grover's algorithm merely halves (leaving ≥128-bit effective security) rather than breaking, on current understanding.

Proof system	Underlying assumption	Quantum-safe?	COGNITUM L2
Groth16 / PLONK	Elliptic-curve pairings	❌ Broken by Shor	Rejected
KZG commitments	Elliptic-curve pairings	❌ Broken by Shor	Rejected
Hash-based STARK (FRI)	Hash collision resistance (SHA-3/SHAKE)	✅ Grover-only (≥128-bit)	**Adopted**

The L1 verifier checks **one STARK proof per batch**, attesting that all *N* L2 transactions in the batch had valid ML-DSA65 signatures, correct per-address nonces, and produced a correct state transition.

16.4 Phase A, Post-Quantum State Channels (Near-Term)

The lowest-complexity L2 and the first to ship:

```
1. Two or more parties lock COGN on L1, opening a channel
2. They exchange ML-DSA65-signed balance updates off-chain (instant, free)
3. Either party can settle the latest signed state on L1 at any time
4. A dispute window lets a counterparty challenge a stale state
5. Channel closes; final balances settle to L1
```

COGNITUM's **single-block BFT finality** (§7.1) makes channel open/close clean, there is no probabilistic reorg risk to wait out. Use cases: streaming payments, gaming, machine-to-machine micropayments.

16.5 Phase B, Post-Quantum Validity Rollup (Medium-Term)

A general-purpose rollup that scales the whole ecosystem, not just bilateral channels:

```
1. A sequencer collects L2 transactions and orders them
2. Transactions execute off-chain against the L2 state
3. A prover generates one hash-based STARK proof for the batch
4. The sequencer posts {proof, compressed state diff} to L1 via the verifier endpoint
5. The L1 verifier checks the single proof and finalizes the batch
```

Because the proof attests to the validity of every transaction in the batch, L1 validators never re-execute the transactions or re-verify the individual ML-DSA65 signatures, they verify one proof. This is what collapses 33.1 MB of signatures (§16.2) into a single on-chain artifact.

16.6 Data Availability and Settlement

The compressed state diff for each batch is posted to L1, so any participant can independently reconstruct full L2 state and exit without the operator's cooperation, the trustless **rollup** mode. A cheaper **validium** mode (state diffs held by a data-availability committee rather than on L1) is available where a weaker availability assumption is acceptable; the tradeoff is documented per-deployment. Settlement reuses the existing transaction path (§5.4) plus a new L1 proof-verifier endpoint.

16.7 Security Inheritance and Trust Model

Property	Source
Post-quantum signatures	Inherited from L1 (ML-DSA65)
Settlement finality	Inherited from L1 (single-block BFT)
Validity of every L2 tx	Enforced by the STARK proof, not by trusting the sequencer
Censorship resistance	Forced-inclusion / escape-hatch: users may submit directly to L1 if the sequencer stalls
Added assumption	Sequencer **liveness** (not safety) and, in validium mode, DA-committee availability

A dishonest sequencer can delay transactions but **cannot forge an invalid state**, an incorrect batch produces a proof the L1 verifier rejects. Users retain an unconditional path to withdraw to L1.

16.8 Roadmap Position

Layer 2 is a **post-mainnet** effort (Phase 5 and beyond). Its prerequisites are precisely the three covered earlier: a **third-party-audited** L1, a **live mainnet**, and **real congestion data** to confirm where scaling is actually needed. Sequence: state channels first (Phase A), then the validity rollup (Phase B).

---

17. Roadmap

Phase 1, Testnet (On-going)

- [x] Post-quantum blockchain core (ML-DSA65)
- [x] BFT consensus + block-scoped PoH
- [x] COGN-20 token standard
- [x] DeFi suite (staking, lending, DEX, bridge)
- [x] Governance system
- [x] KYC and sanctions compliance
- [x] Web dashboard
- [x] Embedded LMDB ledger persistence
- [x] P2P validator network

Phase 2, Mainnet Launch (Q3 2026)

- [ ] Third-party security audit (Certik / Hacken / Quantstamp)
- [ ] Genesis validator recruitment (5-10 independent validators)
- [ ] **External checkpoint anchoring**, periodic block-hash commitments published to an independent notary / public chain, so ledger integrity is verifiable by third parties (externally tamper-evident), not only via the operator's own audit endpoint

- [ ] **Per-block Merkle root + canonical serialization**, compact transaction-inclusion proofs and byte-exact cross-client agreement (replacing whole-block JSON hashing)
- [ ] Mainnet genesis block ceremony
- [ ] Public mainnet launch announcement
- [ ] CoinMarketCap and CoinGecko listings

Phase 3, Exchange Listings (Q3-Q4 2026)

- [ ] Tier 2 CEX listing (Gate.io, MEXC, KuCoin)
- [ ] COGN/USDT and COGN/BTC trading pairs
- [ ] Wrapped COGN (wCOGN) on Ethereum via bridge
- [ ] DEX liquidity bootstrapping

Phase 4, Ecosystem Growth (Q4 2026, Q1 2027)

- [ ] Public COGN-20 token creation (open to all users with COGN)
- [ ] Developer SDK and documentation
- [ ] Developer grant program
- [ ] Mobile wallet (iOS + Android)
- [ ] Tier 1 exchange application (Binance, Coinbase)
- [ ] Institutional partnership program

Phase 5, Post-Quantum Standard (2027)

- [ ] Layer-2 scaling: post-quantum state channels, then a hash-based STARK validity rollup (see §16)
- [ ] SHA-3 hash migration
- [ ] Hardware security module (HSM) validator support
- [ ] Cross-chain post-quantum bridge to other PQC chains
- [ ] COGNITUM as reference implementation for NIST PQC in blockchain

---

18. Ownership and Legal

18.1 Corporate Owner

COGNITUM™ is owned and operated by:

Entity	Role
COGNITUM	Corporate owner, trademark holder, and operating entity, One Person Corporation (SEC-registered, Philippines). A corporate amendment to **COGNITUM BLOCKCHAIN INC.** is in process. **BSP VASP registration and AML/KYC compliance are in process**, and compliance (KYC + sanctions screening) is enforced at the protocol layer (§15). This compliance-first foundation is the basis on which any future, regulated token distribution would be conducted, under qualified securities counsel.
Ryan Paul Gallardo Pillas	Founder, inventor, and principal

18.2 Leadership

Name	Role
Ryan Paul Gallardo Pillas	Chief Executive Officer / Founder

18.3 Intellectual Property

18.4 Risk Disclosure

> This whitepaper is provided for informational purposes. It does not constitute financial advice or an offer of securities. Cryptocurrency investments carry significant risk including total loss of principal. Past performance is not indicative of future results. Regulatory status of COGN may vary by jurisdiction. Prospective participants should consult qualified legal and financial advisors.

Appendix A, API Reference Summary

Endpoint	Method	Description
`/health`	GET	Node health and chain status
`/stats`	GET	Network statistics
`/balance`	GET	COGN balance for address
`/blocks`	GET	Recent blocks
`/tx`	POST	Submit signed transaction
`/keygen`	POST	Generate key pair (body `{scheme}`: `ml-dsa` default, or `slh-dsa`)
`/sign`	POST	Sign message with private key
`/register`	POST	Create account
`/login`	POST	Authenticate
`/token/list`	GET	All COGN-20 tokens
`/token/issue`	POST	Issue new token
`/token/transfer`	POST	Transfer token
`/validator/apply`	POST	Apply as validator
`/governance/propose`	POST	Submit governance proposal
`/governance/vote/:id`	POST	Cast governance vote
`/stake`	POST	Stake COGN
`/lending/lend`	POST	Deposit to lending pool
`/lending/borrow`	POST	Borrow against collateral
`/bridge/swap`	POST	Cross-chain bridge swap
`/swap/swap`	POST	DEX token swap
`/l2/channel/open`	POST	Open a Layer-2 state channel, opener funds their side (phase 1)
`/l2/channel/join`	POST	Party B funds their side, activating the channel (phase 2)
`/l2/channel/cancel`	POST	Cancel an unfunded channel and refund the opener
`/l2/channel/update`	POST	Checkpoint a doubly-signed off-chain balance update
`/l2/channel/close`	POST	Cooperative close, settle final split to L1
`/l2/channel/dispute`	POST	Open a dispute, starting the challenge window
`/l2/channel/challenge`	POST	Submit a higher-version state during a dispute
`/l2/channel/resolve`	POST	Finalize a dispute after its window elapses
`/l2/channels`	GET	List state channels (optionally by party)

Appendix B, Cryptographic Security Parameters

Algorithm	Standard	Security Level	Use Case
ML-DSA65	NIST FIPS 204	192-bit quantum	Transaction signing (default)
SLH-DSA-SHAKE-192f	NIST FIPS 205	192-bit quantum	Transaction signing (optional, hash-based; §6.5)
ML-KEM768	NIST FIPS 203	192-bit quantum	Key encapsulation / encryption
SHA3-256	NIST FIPS 202	128-bit quantum (Grover)	Quantum-safe state/content hash (`QSH-`)
SHA-256	FIPS 180-4	128-bit quantum (Grover)	PoH sequence
AES-256-GCM	FIPS 197	128-bit quantum (Grover)	Symmetric encryption
bcrypt	,	Adaptive	Password hashing
HMAC-SHA256	FIPS 198	Classical	Webhook signature verification
X25519MLKEM768	NIST FIPS 203 (ML-KEM, hybrid)	192-bit quantum (hybrid)	TLS 1.3 transport key exchange (§14.2)